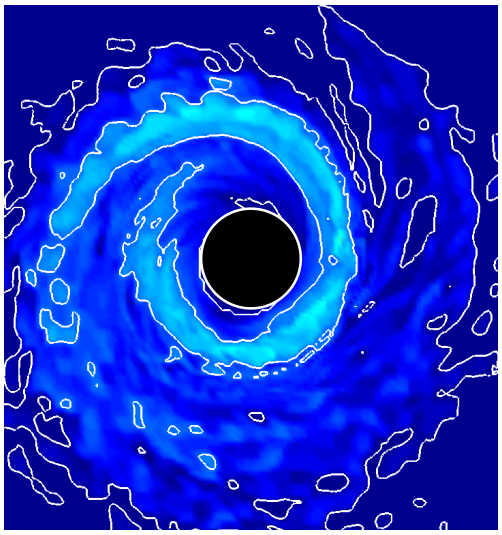
THE UNIVERSITY *of* EDINBURGH

Pseudo-Newtonian Simulations of Black Hole-Neutron Star Mergers as Possible Progenitors of Short-Duration Gamma-Ray Bursts

Hari Sriskantha



Doctor of Philosophy
University of Edinburgh
2014

Abstract

Black hole-neutron star (BH-NS) mergers are promising candidates for the progenitors of short-duration gamma-ray bursts (GRBs). With the right initial conditions, the neutron star becomes tidally disrupted, eventually forming a dense, accreting disk around the black hole. The thermal energy of this black hole-disk system can be extracted via neutrino processes, while the spin energy of the black hole can be extracted via magnetic processes. Either (or even a combination of these) processes could feasibly power a relativistic jet with energy $\gtrsim 10^{49}$ erg and duration $\lesssim 2$ s, hence producing a short-duration GRB.

In this thesis, we investigate BH-NS mergers with three-dimensional, pseudo-Newtonian simulations. We use the simulation code Charybdis, which uses a dimensionally-split, reconstruct-solve-average scheme (i.e. using Riemann solvers) to solve the Euler equations of hydrodynamics. Although the code is based on a Newtonian framework, it includes pseudo-Newtonian approximations of local gravitational wave effects and the innermost stable circular orbit of the BH, which are both general relativistic phenomena. The code also includes the effects of global neutrino emission, shear viscosity and self-gravity.

This thesis comprises two main projects. The first project is a parameter study of the *equation of state*, which encapsulates the relationship between the pressure of a fluid and its other thermodynamic properties. Although the EOS is well understood at low densities, it is yet to be constrained at supranuclear densities, and so must be treated as a parameter in numerical studies of BH-NS mergers. We present simulations using three existing EOSs, in order to investigate their effect on the merger dynamics. We find that the EOS strongly influences the fate of the NS, the properties of the accretion disk, and the neutrino emission.

In the second project, we begin upgrading Charybdis to include *magnetic field effects*, in order to investigate the magnetic processes described above. We implement existing reconstruction and Riemann solver algorithms for the equations of magnetohydrodynamics, and present 1D tests to compare them. When modelling magnetic fields in more than one dimension, we must also deal with the divergence-free condition, $\nabla \cdot \mathbf{B} = 0$. We develop a new constrained transport algorithm to ensure our code maintains this condition, and present 2D tests to confirm its accuracy. This algorithm has many advantages over existing ones, including easier implementation, greater computational efficiency and better parallelisation. Finally, we present preliminary tests that use these algorithms in simulations of BH-NS mergers.

Lay Summary

In the late 1960s, the US military launched a series of satellites to ensure that Soviet Russia were complying with the 1963 Partial Nuclear Test Ban Treaty, which banned the testing of nuclear weapons in the atmosphere. In 1967, these satellites serendipitously started detecting short, intense flashes of gamma rays – a type of high energy light – at a rate of once a day; an astronomical phenomena we now call *gamma-ray bursts* (GRBs). Subsequent satellite missions have given us a lot of information about them: they are the most explosive known astrophysical events, they occur all over the universe, and there are two types of GRBs if we categorise them by how long they last ('short-duration' and 'long-duration').

However, there are still many gaps in our knowledge, such as what causes the short-duration GRBs. The most popular model is the merger of two *compact objects*, such as two neutron stars (NS-NS), or a neutron star and a black hole (BH-NS). (Neutron stars are incredibly dense stars, which are formed when a large, dying star does not have enough mass to collapse all the way into a black hole.) In this thesis, we present computational simulations of such mergers, to investigate whether they could feasibly power a GRB.

Our first project involves the properties of neutron star matter, which are described by an *equation of state* (EOS). We know the properties of low density matter, such as that found on Earth, as they are easy to investigate. However, neutron star matter is unlike anything else: it can reach densities of 10^{15} g cm^{-3}, which is *one quintillion* times denser than lead. This means its properties are yet to be fully determined. Many different EOSs have therefore been proposed, which are based on different physical models. In this work, we present simulations with three such EOSs, to investigate their effect on the outcome of the merger.

Another interesting question is how energy could be extracted from the system to power a GRB. One potential method is through *magnetic processes*, but our simulation code cannot currently model these. Hence, our second project involves upgrading our code to model magnetised fluids. The code can already model non-magnetised fluids, and thankfully, we can use many of the same techniques and algorithms. However, we also have to deal with an added constraint, called the *divergence-free condition*, which says that magnetic fields cannot 'begin' or 'end' anywhere. (This is why diagrams of magnetic field lines always involve closed loops: the loops cannot be broken.) Ignoring this can lead to unphysical results. In this work, we describe a new algorithm to maintain this condition – with many advantages over existing algorithms – and run some tests to confirm its accuracy. Finally, we present some preliminary tests that use our upgraded code in simulations of BH-NS mergers.

Declaration

I declare that this thesis was composed by myself and that the work contained therein is my own, except where explicitly stated otherwise in the text.

(*Hari Sriskantha*)

Acknowledgements

Even though only my name is written on the first page, this thesis would not have been possible without the help and support of many people.

First of all, I would like to thank my supervisor, Maximilian Ruffert, for achieving the perfect balance of making me feel like an independent researcher, while patiently offering all the advice, support and guidance that I needed. I would also like to thank EPSRC for funding this research, and both the School of Mathematics and the Laura Wisewell Travel Fund, for supporting an inspiring and productive conference trip to Kyoto in November 2013.

On a more personal note, I am indebted to friends and colleagues in the School of Mathematics, the Edinburgh Revue and the Musical Medics, for keeping me sane both in and out of the office. In particular, there are a handful of people who deserve special mention, but unfortunately, I am too embarrassed to list their names. They know who they are.

Last but not least, I will always be grateful to my family – especially Zoe – for their continuous love and support, and for listening to me complain endlessly about segmentation faults despite having no idea what that means.

Contents

Part I

Background

Chapter 1
Introduction

1.1 What are Gamma-ray Bursts?

Gamma-ray bursts (GRBs) are transient, astronomical bursts of gamma rays. They are detected about once a day, and last anywhere between a few milliseconds and several minutes. They are the brightest known events since the big bang: their luminosities can be as bright as 10^{52} erg s^{-1}, which means a single GRB is capable of outshining the *rest of the observable universe* in gamma rays. They are also the fastest known events: most bursts have a Lorentz factor $\Gamma \geq 30$, but this has been observed to be as high as 1000 in some cases (Nakar, 2007).

1.1.1 A Brief History

The first GRBs were discovered accidentally in 1967, by the US *Vela* satellites. These were a series of military satellites, which were launched to ensure that Soviet Russia were complying with the 1963 Partial Nuclear Test Ban Treaty. After a delay of six years – due to the classified nature of the data – their discovery was eventually announced by Klebesadel *et al.* (1973). They presented sixteen bursts in the energy range 0.2–1.2 MeV, which lasted from less than a second to around 30 s. They calculated that a source at 1 Mpc would need to emit $\sim 10^{46}$ erg in order to produce the observed responses, and suggested a possible supernova link.

The true nature of these bursts remained mostly enigmatic for over two decades, due to the limitations of gamma-ray instruments of the time. Many theories were proposed; the review paper of Nemiroff (1994) lists over *one hundred possible models* that were published between 1968 and 1992.[1] Although these were initially varied, Nemiroff notes that the research community eventually settled on neutron star progenitors in the galactic disk.

However, these theories were unsettled in 1991 by the launch of NASA's Compton Gamma-Ray Observatory (*CGRO*). This carried the Burst and Transient Source Experiment (*BATSE*), which was specifically designed to detect GRBs and recorded data on 2,704 bursts over its nine-year lifespan. This survey showed that GRBs are isotropically distributed across the sky, with *no* correlation to the galactic plane, suggesting an extra-galactic origin. Additionally, it

[1] Nemiroff (1994) starts their paper by quoting the researcher M. Ruderman: "For theorists who may wish to enter this broad and growing field, I should point out that there are a considerable number of combinations, for example, comets of antimatter following onto white holes, not yet claimed."

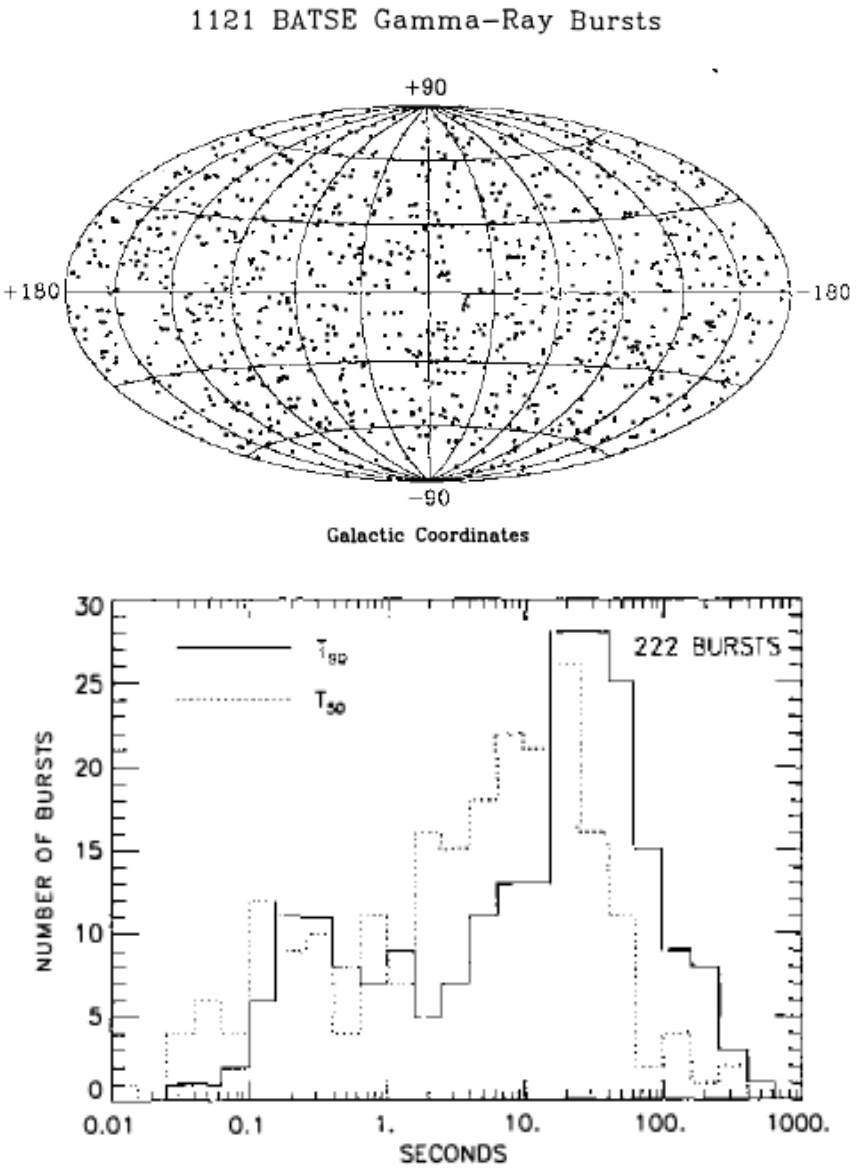


Figure 1.1: Results from *BATSE*, reproduced from Fishman and Meegan (1995). Top: the distribution of 1121 GRBs, plotted in galactic coordinates. There is no correlation to the galactic plane, suggesting GRBs are of extra-galactic origin. Bottom: the duration distribution of 222 GRBs. T_{50} and T_{90} are two different measures, representing the time taken for 50% and 90% of the total burst fluence to be emitted. There appears to be two populations of GRBs, with a division at ~ 2 s.

found that there are (at least) two types of GRBs, if we classify them by their duration (Fishman and Meegan, 1995). Short-duration GRBs (SGRBs) are defined as lasting less than ~ 2 s, while long-duration GRBs (LGRBs) last longer, typically several minutes. The latter are the most common, accounting for approximately 75% of the observed population. Both of these results are shown in Fig. 1.1 (reproduced from Fishman and Meegan, 1995).

The distance to a given GRB cannot be determined via gamma-ray observations alone,

which motivated the search for GRB counterparts at other wavelengths. In 1997, the *Beppo-SAX* satellite – carrying an x-ray telescope – found that the 'prompt' gamma-ray emission of GRB970228 (an LGRB) was followed by a much longer emission in longer wavelengths, which was named the 'afterglow' (Costa *et al.*, 1997). This led to the optical identification of its host galaxy, the first time this had been achieved for a GRB. The redshift of this host was estimated to be $z \sim 0.2 - 2$, providing the first evidence that LGRBs are indeed of cosmological origin (van Paradijs *et al.*, 1997). The afterglows of SGRBs are generally fainter, which meant that they were only detected in 2005 (by the *Swift* and *HETE-2* satellites). These observations confirmed that SGRBs were also of cosmological origin (e.g. Gehrels *et al.*, 2005).

The bimodal classification of GRBs, suggested by the *BATSE* results, is not without problems. For example: defining what is meant by the 'duration' of the GRB is non-trivial; the emission is known to last longer at lower energies, and so detectors operating in different energy bands can measure different durations for the same burst; the division at ~ 2 s is arbitrary, and so there is some overlap between the two groups (e.g. Levan *et al.*, 2007). However, further differences have since been observed between LGRBs and SGRBs (Nakar, 2007):

- SGRBs tend to have harder spectra, as shown in fig. 7 of Fishman and Meegan (1995).
- LGRBs tend to be found in young starbust galaxies; SGRBs can be found in both early- and late-type galaxies, and are associated with older stellar populations.
- LGRBs tend to be found in regions with high star formation rates.
- Nearby LGRBs (with redshift $z < 1$) have been associated with Type Ic supernovae, while no such association has been observed for SGRBs.

These differences are reassuring, as they suggest that SGRBs and LGRBs may have different *progenitors*. We also note that other possible types of GRBs have been suggested, such as 'intermediate-duration bursts' between SGRBs and LGRBs (Zhang, 2011) and 'ultra-long bursts' (Levan *et al.*, 2014). However, the bimodal classification is still the most prevalent. In this thesis, we will focus on the short-duration GRBs.

1.1.2 The Burst Model

The burst model is thought to be the same for both types of GRBs, comprising five stages – as illustrated in Fig. 1.2. The first stage is the **progenitor**, an astrophysical system which undergoes some catastrophic event, creating a **central engine**. This extracts energy from the system, powering the acceleration of an ultra-relativistic jet called the **outflow**. In this thesis, we investigate these three stages: what progenitor and central engine could feasibly produce an outflow that leads to an SGRB? This is discussed further in the next section.

Although this flow starts as optically thick, it becomes optically thin as it cools and spreads out to large distances, $\sim 10^8$ km (Nakar, 2007). The kinetic energy of the flow is then converted to the observed emission via two types of shocks. *Internal* shocks occur when parts of the flow travelling at different speeds collide with each other. This heats the matter and amplifies the gamma-ray emission mechanism, which is thought to be a combination of

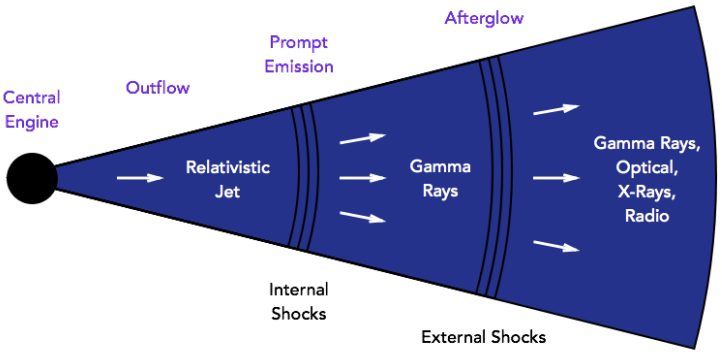


Figure 1.2: Diagram depicting the five stages of a gamma-ray burst.

synchotron radiation and inverse Compton scattering (Rees and Mészáros, 1994). This results in the fourth stage, the observed burst of gamma rays, known as the **prompt emission**. *External* shocks occur when the flow collides with the ambient medium, and likely results in further synchotron emission (Sari *et al.*, 1998). This produces the fifth and final stage, the **afterglow**. Note that we do not investigate these last two stages as part of this thesis.

1.2 Short-Duration GRBs

1.2.1 The Progenitor

The most popular model for the progenitor of SGRBs is the merger of two compact objects in a binary system: either two neutron stars (NS-NS) or a neutron star and a black hole (BH-NS) (as first suggested by e.g. Eichler *et al.*, 1989; Narayan *et al.*, 1992). Based on observational measurements of SGRBs, these mergers are plausible candidates:

- The energy released in a merger is $>\sim 10^{53}$ erg (Narayan *et al.*, 1992). This is sufficient to power a GRB, even if the efficiency of the extraction mechanism is only a few percent.
- Locally, NS-NS mergers are estimated to occur at a rate of 200-3000 $\mathrm{Gpc}^{-3}\ \mathrm{yr}^{-1}$ (Kalogera *et al.*, 2004; Nakar, 2007), while BH-NS mergers are thought to occur 1-10% as frequently (Shibata and Taniguchi, 2011). These are sufficient to explain the observed GRB rates of $\sim \mathrm{day}^{-1}$, even if only a fraction of SGRBs are observed (e.g. due to beaming of the jet).
- The fast variability of the emission ($\gtrsim 10^{-3}$ s) means that – using a causality argument $\Delta x < c\Delta t$ – the progenitor must be smaller than < 100 km (Narayan *et al.*, 1992).
- SGRBs are found in older populations, which are more likely to contain BHs and NSs.

NS-NS systems have been discovered within our galaxy, most famously by Hulse and Taylor (1975), while BH-NS mergers are yet to be observed. In this thesis, we investigate the latter. The black hole and neutron star inspiral and coalesce due to the emission of *gravitational waves*, which carry angular momentum away from the system. If the neutron star becomes tidally disrupted before it is fully absorbed, then it could potentially form a dense, accreting

disk around the black hole; it is this BH-disk system which is thought to form the central engine for SGRBs. Interestingly, such systems are also thought to be the central engine of LGRBs, where the most popular progenitor model is a collapsing massive star. This explains both the association of nearby LGRBs with supernovae and the longer burst duration (as disks created in this manner can be sustained for longer).[2]

We note that compact object mergers are also promising as: (a) sources of gravitational waves that can be detected directly by the next generation of ground-based gravitational-wave detectors, and (b) potential sites for nucleosynthesis via the r-process. However, we will not investigate these research problems as part of this thesis.

1.2.2 The Central Engine

These BH-disk systems are strong candidates for GRB central engines, as they satisfy many of the criteria listed by e.g. Zhang (2011). Firstly, the engine should produce 'clean' jets, with only small amounts of baryon pollution. If there is too much matter, then it would not be possible to accelerate the jet to the high relativistic speeds that have been observed. This problem is neatly solved in BH-NS mergers, because excess baryons can fall through the event horizon of the black hole. Specifically, numerical studies have shown that a *baryon-free funnel* can be created along the rotation axis of the black hole, where material has a low angular momentum, and hence can be absorbed easily (Lee *et al.*, 2000).

Secondly, the engine must be capable of producing the high energies required; if the emission is isotropic, this means energies between 10^{49} and 10^{51} erg. However, these values are reduced if the outflow is beamed. Additionally, the engine must sustain this output for the duration of observed SGRBs. There are two feasible mechanisms for extracting these energies from BH-disk systems: the thermal energy of the disk can be extracted via neutrino processes, while the spin energy of the black hole can be extracted via magnetic processes.

Neutrino Processes

The high temperatures and densities of BH-NS mergers mean that the remnant is mostly opaque to photons, and so the only possible cooling mechanism is via the emission of neutrinos and anti-neutrinos (e.g. Rosswog, 2010). This emission is of interest in the context of GRBs, because some fraction will undergo neutrino-antineutrino annihilation, creating electron-positron pairs, which can subsequently annihilate to produce photons:

$$\nu + \bar{\nu} \rightarrow e^+ + e^- \rightarrow \gamma + \gamma. \tag{1.1}$$

Due to the geometry of the disk, the most efficient $\nu\bar{\nu}$-annihilation occurs along the rotation axis (Nakar, 2007). As mentioned above, there is a baryon-free funnel around this axis, and so it is plausible that pair production here could accelerate a GRB-like relativistic jet. The

[2] BH-disk systems are even known to produce jets in other systems such as active galactic nuclei. However, these cannot be the sources of GRBs, as their accretion rates are too low by several orders of magnitude (Nakar, 2007).

most promising source of neutrino emission for this process is the cooling disk, which means the timescale of this process is tied to the disk lifetime. Numerical studies have shown that it is absorbed within a fraction of a second (e.g. Lee *et al.*, 2005), which is consistent with the durations of SGRBs. Furthermore, a beaming mechanism could be provided by a neutrino-driven wind, which is produced after emitted neutrinos are absorbed by (and hence heat up) other parts of the merger remnant. (The basic idea – as described by Levinson and Eichler (2000) – is that the neutrino-driven wind diverges more slowly than than the streamlines of the relativistic jet, resulting in a collision between the two outflows.)

Neutrino emission from compact object mergers/BH-disk systems have been investigated previously through numerical studies. Ruffert *et al.* (1996) pioneered a 'neutrino leakage scheme', which estimated the neutrino emission by summing the rates of relevant neutrino processes (described further in Section 3.2.1). However, this did not include local effects, such as the neutrino heating described above. Their Newtonian, grid-based investigations of NS-NS mergers found a maximum energy deposition rate from $\nu\bar{\nu}$-annihilation of $\sim 10^{50}$ erg s^{-1}, but found that the disk did not survive long enough to power a GRB.

More recently: Rosswog (2005) ran pseudo-Newtonian, SPH simulations of BH-NS mergers with a GR-mimicking potential, and found a (best-case) maximum energy deposition rate of only $\sim 10^{47}$ erg s^{-1}, which is also insufficient to power a GRB. However, they only investigated high mass ratios $q = M_{BH}/M_{NS}$, which have been shown to produce a lower disk mass (and hence lower the neutrino emission). This effect is described in Chapter 4. Setiawan *et al.* (2006) ran pseudo-Newtonian, grid-based simulations of BH-torus systems (with initial conditions from previous simulations), with an α-model to approximate the shear viscosity. Their results were more optimistic: they found that including viscosity (parameterised by $\alpha = 0.1$) increased the neutrino luminosity by 30-40%, which was enough to feasibly power a SGRB. This was also found by Deaton *et al.* (2013), who presented the first relativistic simulations of BH-NS mergers with a realistic equation of state and a neutrino treatment. However, it is important to note that the neutrino emission is sensitive to a number of parameters – including, as will be discussed in Chapter 4, the equation of state – so these results cannot be compared directly with each other. Instead, they contribute to our wider understanding.

Magnetic Processes

A limitation of neutrino processes is that they only tap the thermal energy of the disk. However, there is another energy source – the spin energy of the black hole – which can be tapped through *magnetic processes*. Even if the initial magnetic field is weak, it will be amplified by the magnetorotational (Nakar, 2007) and Kelvin-Helmholtz instabilities (Price and Rosswog, 2006). The timescales of these processes are also tied to the accretion disk – which is required to stop the magnetic field from dispersing – and are therefore still consistent with the durations of SGRBs (Lee *et al.*, 2000). Additionally, a poloidial magnetic field threading the BH will both: (a) provide a barrier against charged baryons, reducing the baryonic pollution and (b) relax the constraint on the amount of baryon pollution permitted (Lei *et al.*, 2013).

One such magnetic process is the Blandford-Znajek mechanism, where magnetic braking generates a Poynting flux along the low-baryon funnel (Blandford and Znajek, 1977). The energy released is proportional to the square of the magnetic field strength ($\mathbf{B}^2$). Subsequently, it can be shown that a field strength of $\mathbf{B} \sim 10^{15}$ G is sufficient to power a high-energy GRB with $\sim 10^{53}$ erg (Lee *et al.*, 2000). Additionally, this energy 'increases sharply' with the spin of the BH, which means this process can be more efficient that a neutrino-powered jet (Nakar, 2007). Another possible process is the Blandford-Payne mechanism, where the magnetic field drives outflows from the accretion disk (Blandford and Payne, 1982).

Again, magnetic fields from compact object mergers/BH-disk systems have been investigated previously through numerical studies. Price and Rosswog (2006) ran Newtonian SPH simulations of NS-NS mergers, which found that the magnetic field amplifies on the timescale of ~ 1 ms due to the Kelvin-Helmoltz instability, and could feasibly power a jet of energy 10^{51} erg. Rezzolla *et al.* (2011) ran relativistic simulations of NS-NS mergers, and found that an initial field of 10^{12} G was amplified to produce a poloidal field of 10^{15} G, leading to jet-like structures. Etienne *et al.* (2012) ran relativistic simulations of BH-NS mergers, and found that when a NS was seeded with a strong magnetic field ($\sim 10^{17}$ G), the final mass of the disk was increased by up to 100% compared to the non-magnetised case. Although they do not analyse any potential jets, this will enhance the neutrino emission, which increases with disk mass. Shibata and Sekiguchi (2012) ran relativistic simulations of BH-torus systems, with neutrino emission (but without a fully-realistic EOS). They found energy deposition from neutrino-antineutrino annhiliation to be only 10^{48} erg, but they point out that this is a lower bound due to microphysics limitations. They also do not analyse any magnetic-driven jets.

1.3 Structure of this Thesis

In this thesis, we present and develop three-dimensional, pseudo-Newtonian simulations of BH-NS mergers, to investigate whether they are plausible progenitors for short-duration GRBs. In the rest of Part I, we describe our simulation code, Charybdis. This is built on a hydrodynamics engine, which uses a dimensionally-split, reconstruct-solve-average scheme (i.e. using Riemann solvers) to solve the Euler equations of hydrodynamics. These methods are summarised in Chapter 2. In Chapter 3, we provide a brief summary of the numerical methods used to model additional physics, including neutrino emission, viscosity, self-gravity and gravitational waves. We also describe the black hole and neutron star models used.

In Part II, we present the results of our investigations. These are split into two main projects, which are summarised below. The first project involves the equation of state, and is described in Chapter 4. The second project involves the inclusion of magnetic field effects in Charybdis, and is described in Chapters 5-7. Finally, in Chapter 8, we conclude with a summary of results, and thoughts on possible future directions for this work.

1.3.1 Project One: the Equation of State

In the first project, we present a parameter study of the *equation of state* (EOS), which encapsulates the relationship between the pressure of a fluid and its other properties, such as its mass density, temperature and composition. Although the neutron star EOS is well understood at low densities, it is yet to be constrained at supranuclear densities (that is, above the nuclear saturation density, $\rho \sim 10^{14.7}$ g cm^{-3}). Consequently, the EOS must be treated as a parameter in numerical studies of BH-NS mergers. In Chapter 4, we present BH-NS merger simulations with three realistic EOSs derived from nuclear theory, by H. Shen *et al.* (1998a b), G. Shen *et al.* (2011a) and G. Shen *et al.* (2011b). Our aim is to investigate the effect of the EOS on the merger dynamics and the disk. We also discuss the effect on the neutrino emission, which – as described above - is important in the context of neutrino-driven GRB production.

1.3.2 Project Two: Magnetic Fields

In the second project, we begin upgrading Charybdis to include the effect of *magnetic fields*, with the long-term goal of investigating the magnetic processes described above. At the time of writing, no merger simulations have been presented with both magnetic fields *and* a neutrino treatment with a realistic EOS, and so this is a goal worth pursuing.

To do this, we must upgrade the hydrodynamics engine to solve the equations of *magneto*hydrodynamics (MHD), which combine the Euler equations with Maxwell's equations of electromagnetics. In Chapter 5, we implement reconstruction and Riemann solver algorithms for the equations of magnetohydrodynamics, and present 1D tests to compare them. When modelling magnetic fields in two or more dimensions, we must also deal with the divergence-free condition, $\nabla \cdot \mathbf{B} = 0$. In Chapter 6, we develop a new constrained transport algorithm to ensure our code maintains this condition, and present 2D tests to confirm its accuracy. This algorithm has many advantages over existing ones, including easier implementation, greater computational efficiency and better parallelisation. This work has been published in the *Monthly Notices of the Royal Astronomical Society* (Sriskantha and Ruffert, 2013); for reference, a copy of the submitted paper is provided in Appendix C. Finally, in Chapter 7, we run a number of preliminary tests that use these algorithms in simulations of BH-NS mergers.

Chapter 2
Numerical Hydrodynamics

To simulate black hole-neutron star mergers, we need to model many physical effects. We start with the flow of fluids, which are modelled by the Euler equations of hydrodynamics. In this chapter, we show the derivation of these equations, as well as the methods used by our simulation code, Charybdis, to solve them numerically. Note that these were not implemented as part of this thesis.

2.1 Conservation Laws

2.1.1 Derivation

We start with the derivation of a generic 1D conservation law, as described in e.g. LeVeque (1992). Consider some physical quantity of the fluid in terms of its *density*, $u(x,t)$, such as the mass density or the total energy density. The 'amount' of this quantity in the one-dimensional interval $[x_1, x_2]$ can be obtained by integrating this density over space:

$$\int_{x_1}^{x_2} u(x,t)dx. \tag{2.1}$$

We also define the *flux* of this quantity as some function $f(u(x,t))$, which is the amount of the quantity flowing past the point x at time t. Hence, if the quantity is *conserved*:

$$\frac{d}{dt}\int_{x_1}^{x_2} u(x,t)dx = f(u(x_1,t)) - f(u(x_2,t)). \tag{2.2}$$

This is physically intuitive: the rate of change of the amount of the quantity within $[x_1, x_2]$ must be equal to the flux entering at x_1 minus the flux leaving at x_2. Note that we have defined a positive flow as travelling from x_1 to x_2. And so to determine the *total* change in this quantity between two times – say t_1 and t_2 – we can integrate eq. (2.2) in time:

$$\int_{x_1}^{x_2} u(x,t_2)dx = \int_{x_1}^{x_2} u(x,t_1)dx + \int_{t_1}^{t_2} f(u(x_1,t))dt - \int_{t_1}^{t_2} f(u(x_2,t))dt, \tag{2.3}$$

where, for example, the integral $\int_{t_1}^{t_2} f(u(x_1,t))dt$ represents the total flux flowing past x_1 during the time interval $[t_1, t_2]$. This is the *integral form* of the 1D conservation law.

Alternatively, if we assume that f is differentiable, then:

$$f(u(x_2,t)) - f(u(x_1,t)) = \int_{x_1}^{x_2} \frac{\partial}{\partial x} f(u(x,t))dx. \tag{2.4}$$

We can then rearrange eq. (2.2) to give:

$$\int_{x_1}^{x_2} \frac{\partial}{\partial t} u(x,t)dx + \int_{x_1}^{x_2} \frac{\partial}{\partial x} f(u(x,t))dx = 0. \tag{2.5}$$

If we group the integrals together, we get:

$$\int_{x_1}^{x_2} \left\{ \frac{\partial}{\partial t} u(x,t) + \frac{\partial}{\partial x} f(u(x,t)) \right\} dx = 0. \tag{2.6}$$

Since this must be true for any interval $[x_1, x_2]$, then the integrand must always equal zero. Hence, we obtain the *differential form* of the 1D conservation law:

$$\frac{\partial}{\partial t} u(x,t) + \frac{\partial}{\partial x} f(u(x,t)) = 0. \tag{2.7}$$

We aim to solve this conservation law, given some initial condition, in order to determine how the fluids we are modelling evolve with time. However, finding analytic solutions for systems of physical interest – such as black hole–neutron star mergers – is a difficult task, if not impossible, especially if we wish to investigate many different plausible scenarios. A more practical alternative, therefore, is to take a *numerical* approach.

2.1.2 Discretisation and Notation

Before we can do this, we must first discretise the system we wish to model. We move to 3D Cartesian space in the domain $D_x \times D_y \times D_z$. This space is divided into $N_x \times N_y \times N_z$ 'cells' in the usual way, each of volume $\Delta x \times \Delta y \times \Delta z$, such that $D_x = N_x\, \Delta x$, etc. For most of the algorithms described in this thesis, we will assume that these cell sizes are constant and uniform.[1] We then refer to this computational grid using the following conventions:

- Each cell is centred at position (x_i, y_j, z_k), where $i \in \{1, 2, ..., N_x\}$ and $x_i = (i - 1/2)\, \Delta x$, etc. We refer to the cell centred at position (x_i, y_j, z_k) using the co-ordinates (i, j, k).
- We refer to cell faces and cell edges using half-indices. For brevity, we use starred notation, such that $i* = i + 1/2$. A face will therefore have one starred co-ordinate, e.g. $(i*, j, k)$, while an edge will have two, e.g. $(i*, j*, k)$.
- We discretise time such that $t_n = \sum_{m=1}^{n} \Delta t_m$. The timestep Δt_m can change in time, but must always satisfy the Courant-Friedrich-Lewy (CFL) stability condition.[2]

A typical cell, face and edge are depicted in Fig. 2.1.

[1] In reality, our simulation code Charybdis uses a nested grid scheme to solve the problem of modelling different length scales. The implementation of this is described in Ruffert (1992), but is also briefly discussed in Section 2.5.

[2] The CFL stability condition is also briefly discussed in Section 2.5.

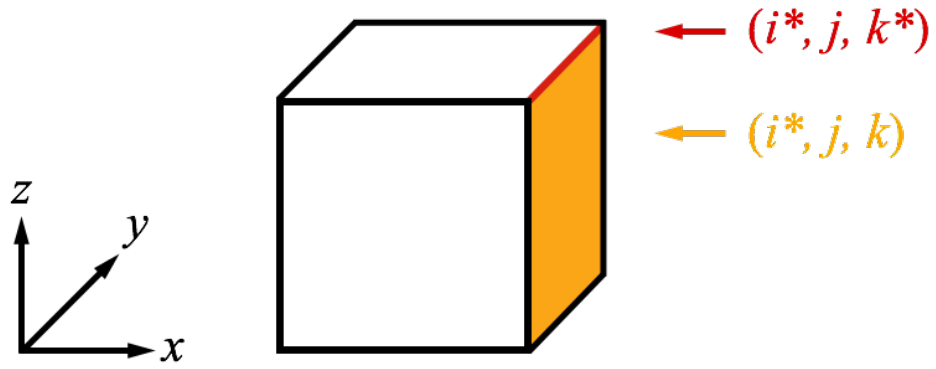


Figure 2.1: This figure depicts the cell (i, j, k). Using the notation described in section 2.1.2, the orange face is labelled $(i*, j, k)$ – where $i* = i + 1/2$ – and the red edge is labelled $(i*, j, k*)$.

Finally, we must discretise the data itself. Using this notation, we can now define the *cell average* of the density u, which is a spatial average over the cell (i, j, k) at time t_n:

$$U_n^{i,j,k} = \frac{1}{\Delta x \Delta y \Delta z} \int_{z_{k*-1}}^{z_{k*}} \int_{y_{j*-1}}^{y_{j*}} \int_{x_{i*-1}}^{x_{i*}} u(x, y, z, t_n) dx dy dz, \tag{2.8}$$

where we use a capital U for the discretised cell averages. We then store and evolve these cell averages, rather than the original function $u(x, y, z, t_n)$. Similarly, we can define a *face average* of the flux f, which is an average over the face $(i*, j, k)$ *and* the time interval $[t_n, t_{n+1}]$:

$$F_{n*}^{i*,j,k} = \frac{1}{\Delta t \Delta y \Delta z} \int_{t_n}^{t_{n+1}} \int_{z_{k*-1}}^{z_{k*}} \int_{y_{j*-1}}^{y_{j*}} f(x_{i*}, y, z, t) dx dy dt. \tag{2.9}$$

Note that we follow the convention of referring to fluxes at the half timestep, $n* = n + 1/2$.

To avoid excessive indices, we only write down spatial co-ordinates that *differ* from (i, j, k). For example, the cell $(i, j + 1, k)$ will be abbreviated to $(j + 1)$. Hence the cell average of the quantity u on the cell $(i, j + 1, k)$ at time t_n is just written as: U_n^{j+1}. The reader should therefore assume that there are always as many spatial co-ordinates as the number of dimensions we are working in, even if all of them are not explicitly written down. When there is no change from (i, j, k), we use $\bullet$, such that $U_n^{i,j,k}$ is abbreviated to $U_n^{\bullet}$. Furthermore, we do not use subscripts to denote components of discretised vectors. For example: the cell-average of the x-component of the velocity will be written as $(vx)_n^{\bullet}$, not $v_{x,n}^{\bullet}$. A summary of the notation used throughout this thesis is provided in the 'Summary of Notation' on page viii.

2.1.3 Finding Numerical Solutions

Given that conservation laws can be written in differential form, it is plausible that we could find solutions using *finite difference* schemes, which use numerical approximations to the partial derivatives. However, this approach does not generally work near discontinuities,

where the differential form no longer holds: we assume u and f are smooth and differentiable in order to obtain the differential form in eq. (2.7) from the integral form in eq. (2.3). This will cause problems in astrophysical scenarios, where discontinuities such as shock waves are common (and can even arise from conditions which were initially smooth).

One possible solution is *shock tracking*, where discontinuities are explicitly tracked and dealt with, and finite differences are only used in sufficiently smooth regions of the flow. However, this can result in inelegant and complex algorithms. Instead, a preferable approach is *shock capturing*, in which discontinuities are dealt with automatically. Most modern shock capturing methods are based on the *reconstruct-solve-average* strategy (RSA), so called for the three stages involved.[3] This is based on the more general, integral form of the conservation law, which *does* hold at discontinuities. To see how this works, we start by describing Godunov's scheme: the simplest RSA strategy, and indeed the pioneer of this family of schemes.

2.2 Godunov's Scheme

2.2.1 Reconstruct-Solve-Average

We go back to considering a 1D system, with some conserved quantity u. We start with the cell-averaged value of u in every cell, which – in one dimension – is defined as:

$$U_n^i = \frac{1}{\Delta x} \int_{x_{i*-1}}^{x_{i*}} u(x, t_n) dx. \tag{2.10}$$

The first step of the RSA strategy is to use these cell averages to **reconstruct** a piecewise continuous function $\tilde{u}_n(x,t)$ that is some approximation of the actual function $u(x,t_n)$. Godunov's scheme uses the following first-order interpolation (Godunov, 1959):

$$\tilde{u}_n(x,t_n) = \sum_i \chi_i(x)\, U_n^i \qquad \text{where } \chi_i(x) = \begin{cases} 1 & \text{if } x_{i*-1} < x < x_{i*} \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

Here, the function $\tilde{u}_n$ takes the value U_n^i everywhere within the ith cell, as depicted in Fig. 2.2. The main purpose of this reconstruction is that it allows us to estimate the value of $u(x,t_n)$ on either side of the cell interface $i*$; we label these as U_L^{i*} (left state) and U_R^{i*} (right state) respectively. Using eq. (2.11) – and as shown in Fig. 2.2 – these are clearly given by:

$$U_L^{i*} = U_n^i, \qquad U_R^{i*} = U_n^{i+1}. \tag{2.12}$$

We can envisage non-smooth flows in which $U_L^{i*} \neq U_R^{i*}$, in which case this reconstruction defines a *Riemann problem* at the cell interface $i*$: the solution to a conservation law with piecewise initial data and a single discontinuity (LeVeque, 1992).

Hence we arrive at the major advantage of this approach: it is possible to **solve** each Rie-

[3] Note that this is also sometimes referred to as the reconstruct-*evolve*-average strategy.

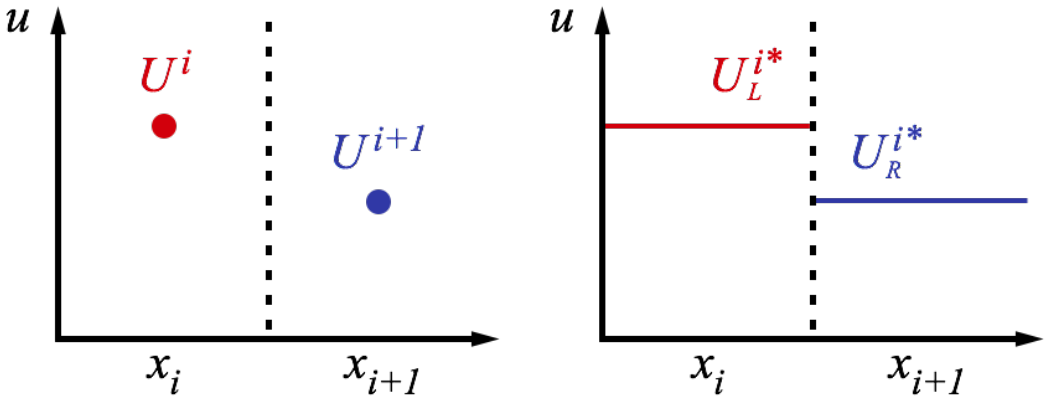


Figure 2.2: *Reconstruction*: the left-hand figure shows some typical discretised data. The right-hand figure shows a first-order interpolation of this initial data, where the reconstructed function takes the value U_n^i in the interval $[x_{i*-1}, x_{i*}]$. This is Godunov's scheme, as described in Section 2.2. Note that this defines a *Riemann problem* at the cell-interface: a conservation law with piecewise-constant initial data and a single discontinuity. We label the 'input states' to the Riemann problem as U_L^{i*} and U_R^{i*}.

mann problem exactly, to obtain solutions at time t_{n+1}, as long as the timestep $\Delta t = t_{n+1} - t_n$ is sufficiently small to prevent neighbouring Riemann problems from interacting with each other. (This is in addition to the CFL condition mentioned in section 2.1.2.) A detailed explanation of how Riemann solver algorithms work is outside the scope of this thesis, but can be found in e.g. LeVeque (1992). A brief summary for hydrodynamics is given in Section 2.4.4.

The final stage is to combine these solutions to get an approximation to the function $u(x, t_{n+1})$, which we call $\tilde{u}_n(x, t_{n+1})$. This is another reason why we must observe the timestep condition described above: if neighbouring Riemann problems do not interact with each other, then we can simply piece the solutions for each cell interface together. We then re-discretise this data by taking the **average** of this new function over each cell:

$$U_{n+1}^i = \frac{1}{\Delta x} \int_{x_{i*-1}}^{x_{i*}} \tilde{u}_n(x, t_{n+1}) dx. \tag{2.13}$$

The system has now been evolved for one timestep.

2.2.2 Simplifying Godunov's Scheme

We can simplify this procedure by considering the integral form of the 1D conservation law given in eq. (2.3). If we define the temporal 1D flux average as:

$$F_{n*}^{i*} = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} f(x*, t) dt, \tag{2.14}$$

then we can rewrite eq. (2.3) in terms of cell and flux averages:

$$U^i_{n+1} = U^i_n - \frac{\Delta t}{\Delta x}\left[F^{i*}_{n*} - F^{i*-1}_{n*}\right]. \tag{2.15}$$

Note that this equation is still exact. This is a neat way of thinking about the conservation law in terms of discretised data: the problem of finding the value U^i_{n+1} is reduced to the problem of finding the *intercell fluxes* F^{i*}_{n*}, which we obtain from the Riemann solver during the solve stage. We can then replace the average stage with an update defined by eq. (2.15). Since these fluxes depend on the left and right states U^{i*}_L and U^{i*}_R, we can rewrite this as:

$$U^i_{n+1} = U^i_n - \frac{\Delta t}{\Delta x}\left[F^{i*}_{n*}(U^{i*}_L, U^{i*}_R) - F^{i*-1}_{n*}(U^{i*-1}_L, U^{i*-1}_R)\right]. \tag{2.16}$$

More generally, these left and right states depend on the cell averages, and so:

$$U^i_{n+1} = U^i_n - \frac{\Delta t}{\Delta x}\left[F^{i*}_{n*}(U^i_n, U^{i+1}_n) - F^{i*-1}_{n*}(U^{i-1}_n, U^i_n)\right]. \tag{2.17}$$

This equation encapsulates the entirety of Godunov's method, and indeed all RSA schemes.

To summarise the RSA strategy: for each timestep, we reconstruct the cell-averaged data U^i_n to estimate the values of the original function $u(x, t_n)$ on either side of each cell interface $i*$, which we call U^{i*}_L and U^{i*}_R. This defines a sequence of Riemann problems. We can solve each of these problems independently using a Riemann solver algorithm (for a sufficiently small timestep), which returns the flux F^{i*}_{n*} through the cell interface $i*$. We then use these intercell fluxes F^{i*}_{n*} to determine the new cell-averaged state quantities, U^i_{n+1}, using eq. (2.17).

2.3 Hydrodynamics and the Euler Equations

2.3.1 Dimensional Splitting

Now that we have described the RSA strategy for a 1D system, the next step is to extend it to all three spatial dimensions. The differential form of the conservation law becomes:

$$\frac{\partial}{\partial t}u(x,y,z,t) + \frac{\partial}{\partial x}f(u(x,y,z,t)) + \frac{\partial}{\partial y}g(u(x,y,z,t)) + \frac{\partial}{\partial z}h(u(x,y,z,t)) = 0, \tag{2.18}$$

where g and h are the flux functions in the y- and z-directions respectively. The integral/discretised form can be extended in a similar way to become:

$$U^\bullet_{n+1} = U^\bullet_n - \frac{\Delta t}{\Delta x}\left[F^{i*}_{n*} - F^{i*-1}_{n*}\right] - \frac{\Delta t}{\Delta y}\left[G^{j*}_{n*} - G^{j*-1}_{n*}\right] - \frac{\Delta t}{\Delta z}\left[H^{k*}_{n*} - H^{k*-1}_{n*}\right]. \tag{2.19}$$

Although three-dimensional Riemann solvers have been developed, which simultaneously find the intercell fluxes through all six faces of the 3D cell, these are computationally expen-

sive. Instead, a popular alternative is the *dimensionally-split* (DS) approach:

$$U_1^\bullet = U_n^\bullet - \frac{\Delta t}{\Delta x}\left[F_{n*}^{i*} - F_{n*}^{i*-1}\right], \tag{2.20}$$

$$U_2^\bullet = U_1^\bullet - \frac{\Delta t}{\Delta y}\left[G_{n*}^{j*} - G_{n*}^{j*-1}\right], \tag{2.21}$$

$$U_{n+1}^\bullet = U_2^\bullet - \frac{\Delta t}{\Delta z}\left[H_{n*}^{k*} - H_{n*}^{k*-1}\right]. \tag{2.22}$$

In other words, each timestep is broken down into an x-, y- and z-'direction pass', represented by eqs. (2.20), (2.21) and (2.22) respectively. In each of these passes, the 3D grid is split into several independent 1D problems along the direction of the pass. For example, in the x-direction pass, there are $N_y \times N_z$ 'rows' that are each defined by fixed j and k. We can evolve each of these rows independently according to the 1D RSA strategy described in Section 2.2, in order to determine the cell average $U_1^\bullet$ in all cells using eq. (2.20). We then use these values as the initial conditions for the y-direction pass, and so on.

The popularity of this approach is due to its many advantages: the timestep is not as constricted by the CFL stability condition as with a fully multi-dimensional code; it is simple to upgrade an existing 1D code to more dimensions in this way, as we do not need to re-code the RSA algorithm; the rows in each pass can be evolved independently, which means the code can be easily parallelised; it can be shown that the accuracy of the 1D code can be preserved by alternating the order of the passes (Liska and Wendroff, 2003).

2.3.2 The Euler Equations

We are now ready to apply the algorithms and methods described so far to the Euler equations of hydrodynamics. These are a set of hyperbolic, partial differential equations, used to model the flow of compressible fluids, such as those found in many astrophysical scenarios.

The conservation law described in Section 2.1 is applied to five densities of interest: the mass density ρ, the linear momentum density $\rho\mathbf{u} = (\rho u_x, \rho u_y, \rho u_z)$, and the total energy density $\mathcal{E}$. These *state quantities* can be encapsulated in a *state vector*, $\mathbf{u} = \mathbf{u}(x,t)$, such that:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} + \frac{\partial \mathbf{g}(\mathbf{u})}{\partial y} + \frac{\partial \mathbf{h}(\mathbf{u})}{\partial z} = 0, \tag{2.23}$$

where $\mathbf{f}(\mathbf{u})$, $\mathbf{g}(\mathbf{u})$ and $\mathbf{h}(\mathbf{u})$ are *flux vectors*, containing the corresponding flux functions. These are known analytically, so we can write the Euler equations in full as (LeVeque, 1992):

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho v_x\\ \rho v_y\\ \rho v_z\\ \mathcal{E}\end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix}\rho v_x\\ \rho v_x^2 + p\\ \rho v_x v_y\\ \rho v_x v_z\\ v_x(\mathcal{E}+p)\end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix}\rho v_y\\ \rho v_x v_y\\ \rho v_y^2 + p\\ \rho v_y v_z\\ v_y(\mathcal{E}+p)\end{bmatrix} + \frac{\partial}{\partial z}\begin{bmatrix}\rho v_z\\ \rho v_x v_z\\ \rho v_y v_z\\ \rho v_z^2 + p\\ v_z(\mathcal{E}+p)\end{bmatrix} = 0. \tag{2.24}$$

Direction pass:	**x**	**y**	**z**
$(\rho vx)^{1D}$ stores	(ρvx)	(ρvy)	(ρvz)
$(\rho vy)^{1D}$ stores	(ρvy)	(ρvx)	(ρvx)
$(\rho vz)^{1D}$ stores	(ρvz)	(ρvz)	(ρvy)

Table 2.1: In a dimensionally-split code, we can avoid rewriting the 1D algorithms if we make sure that the Cartesian components of the linear momentum are extracted (and returned) according to the scheme described in this table. We will also use this strategy for the components of the magnetic field.

where p is the gas pressure. Note that we have six unknown quantities (ρ, ρv_x, ρv_y, ρv_z, E and p), but only five equations. Hence, we require another equation to close the system before we can find unique solutions. This is the *equation of state* (EOS), which relates the gas pressure to other thermodynamic quantities. A simple example is the ideal gas law:

$$\epsilon = \frac{p}{\rho(\gamma - 1)}, \tag{2.25}$$

where ϵ is the internal energy, such that $\mathcal{E} = \rho\epsilon + \frac{1}{2}\rho|\mathbf{v}|^2$ is the total energy and $\frac{1}{2}\rho|\mathbf{v}|^2$ is the kinetic energy. Unsurprisingly, this is a very poor physical approximation – especially for the extreme conditions found in BH-NS mergers – and so we would prefer a more 'realistic' EOS. More details about the approach we use are given in Chapter 4.

So now we have five versions of eq. (2.19): one for each state quantity. One subtle detail is that the discretised values of the linear momentum – which are stored as Cartesian components, $(\rho vx)^{\bullet}_n$, $(\rho vy)^{\bullet}_n$ and $(\rho vz)^{\bullet}_n$ – must be extracted differently for each direction pass. For example, in the x-direction pass, the component $(\rho vx)^{\bullet}_n$ is parallel to the row, and is extracted and stored in a 1D row we shall call $(\rho vx)^{1D}$. However, in the y-direction pass, it is the $(\rho vy)^{\bullet}_n$ component that is parallel to the row. To avoid having to rewrite the 1D algorithms for each directional pass, we simply store $(\rho vy)^{\bullet}_n$ in the array $(\rho vx)^{1D}$, and similarly switch the other components. This approach is summarised in Table 2.1.

2.4 Improving Godunov's Scheme

There are a number of modifications we can make to Godunov's scheme, to improve its accuracy and efficiency. In order to understand how these work, we start by summarising some of the mathematical properties of hyperbolic PDEs (e.g. LeVeque, 2002).

2.4.1 Hyperbolic PDEs

Consider a general linear system of m 1D conservation laws:

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = 0, \tag{2.26}$$

where $\mathbf{u}(x,t)$ is a vector of conserved quantities, $\mathbf{f}(\mathbf{u})$ is the vector of corresponding fluxes and for brevity, we use the notation u_x to represent the partial derivative $\partial u/\partial x$. Using the $m \times m$ flux Jacobian matrix $\mathbf{A} = \mathbf{f_u}$, we can rewrite eq. (2.26) as:

$$\mathbf{u}_t + \mathbf{A}\mathbf{u}_x = 0. \tag{2.27}$$

This system is classed as *hyperbolic* if $\mathbf{A}$ is diagonalisable with m real eigenvalues, $\lambda_1, \lambda_2, ...\lambda_m$ and corresponding (right) eigenvectors $\mathbf{r}_1, \mathbf{r}_2, ...\mathbf{r}_m$, such that:

$$\mathbf{\Lambda} = \mathbf{R}^{-1}\mathbf{A}\mathbf{R} = \mathbf{L}\mathbf{A}\mathbf{R}, \tag{2.28}$$

where the matrix $\mathbf{R}$ and the diagonal matrix $\mathbf{\Lambda}$ are defined as:

$$\mathbf{R} = \left[\begin{array}{c|c|c} & & \\ \mathbf{r}_1 & ... & \mathbf{r}_m \\ & & \end{array}\right] \quad \text{and} \quad \mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & & \\ & ... & \\ & & \lambda_m \end{bmatrix}, \tag{2.29}$$

and $\mathbf{L} = \mathbf{R}^{-1}$ is the corresponding matrix of left eigenvectors.

We can premultiply eq. (2.27) by the matrix $\mathbf{L}$, to get the relation:

$$\mathbf{L}\mathbf{u}_t + \mathbf{L}\mathbf{A}\mathbf{u}_x = 0. \tag{2.30}$$

Additionally, we can arbitrarily add the term $\mathbf{RL}$, which is equal to the identity matrix:

$$\mathbf{L}\mathbf{u}_t + \mathbf{L}\mathbf{A}\mathbf{R}\mathbf{L}\mathbf{u}_x = 0. \tag{2.31}$$

If we then define a new vector of variables, $\mathbf{w}(x,t) = \mathbf{L}\mathbf{u}(x,t)$, we are left with:

$$\mathbf{w}_t + \mathbf{\Lambda}\mathbf{w}_x = 0. \tag{2.32}$$

In other words: we have reframed eq. (2.26) in terms of m independent advection equations. Each element w^p of the vector $\mathbf{w}$ – known as a *characteristic variable* – propagates as a wave travelling at speed λ^p. The trajectory of the p^{th} wave from some point x_0 is called the *characteristic curve* or p-characteristic, and is defined as $\chi_p(t) = x_0 + \lambda_p t$ (a straight-line for constant λ^p). Hence, if we know $w^p(x,0)$ at time $t = 0$, then $w^p(x,\tau)$ at time $t = \tau$ is given by:

$$w^p(x,\tau) = w^p(x - \lambda^p\tau, 0). \tag{2.33}$$

The solution in terms of the conserved variables $\mathbf{u}$ is then given by:

$$\mathbf{u}(x,t) = \mathbf{w}(x,t)\mathbf{R} = \sum_{p=1}^{m} w^p(x,t)r^p. \tag{2.34}$$

Because the system is hyperbolic, this decomposition of $\mathbf{u}$ is unique.

2.4.2 Riemann Problems

It is straightforward to extend this framework to find the solution of a Riemann problem. Recall that our initial data now includes a discontinuity (say, at $x = 0$), such that:

$$\mathbf{u}(x,0) = \begin{cases} \mathbf{u}_l & \text{if } x < 0, \\ \mathbf{u}_r & \text{if } x > 0. \end{cases} \tag{2.35}$$

This can be transformed to characteristic variables in the usual way:

$$w^p(x,0) = \begin{cases} w_l^p & \text{if } x < 0, \\ w_r^p & \text{if } x > 0. \end{cases} \tag{2.36}$$

Using eq. (2.33), we can find $w^p(x,\tau)$ at time $t = \tau$:

$$w^p(x,\tau) = \begin{cases} w_l^p & \text{if } x - \lambda^p t < 0, \\ w_r^p & \text{if } x - \lambda^p t > 0. \end{cases} \tag{2.37}$$

If we then define $P(x,t)$ as the maximum value of p for which $x - \lambda^p t > 0$, then the solution in terms of the conserved variables $\mathbf{u}$ is given by:

$$\mathbf{u}(x,t) = \sum_{p=1}^{P(x,t)} w_r^p r^p + \sum_{p=P(x,t)+1}^{m} w_l^p r^p \tag{2.38}$$

In other words: a discontinuity in the conserved variables at the point $x_0 = 0$ results in a discontinuity in one or more of the characteristic variables at this point. These discontinuities then propagate along the corresponding characteristic curves, leading to discontinuities in the solution $\mathbf{u}(x,t)$ at some or all of the points $x_0 + \lambda^p t$ (LeVeque, 2002).

2.4.3 The Euler Equations

In the case of the 1D Euler equations, there are $m = 3$ equations, and hence three characteristic waves. The three eigenvalues are $\lambda_1 = v - c$, $\lambda_2 = v$, $\lambda_3 = v + c$, where c is the sound speed. Note that we are now working in the *primitive variables* (ρ, v, p), instead of the conserved variables $(\rho, \rho v, \mathcal{E})$. We can switch between the two using the equation of state.

The discussion so far has focused on a linear system of equations. However, the Euler equations are *nonlinear*: there are still $m = 3$ characteristic waves, but these can become compressed or expanded. In the case of a Riemann problem, this results in a *shock wave* (a simple discontinuity) or a *rarefaction wave* (a wave with a head and a tail which travel at different speeds, resulting in a spread-out rarefaction fan). In Fig. 2.3, the 1- and 3-waves can be either of these two waves, causing a jump in ρ, v and p. The central 2-wave is just a standard linear discontinuity – often called a *contact discontinuity* – which causes a jump in ρ only. There is no pressure gradient to move the fluid from one side of the contact discontinuity to the other, and so it simply propagates at the speed $\lambda_2 = v$.

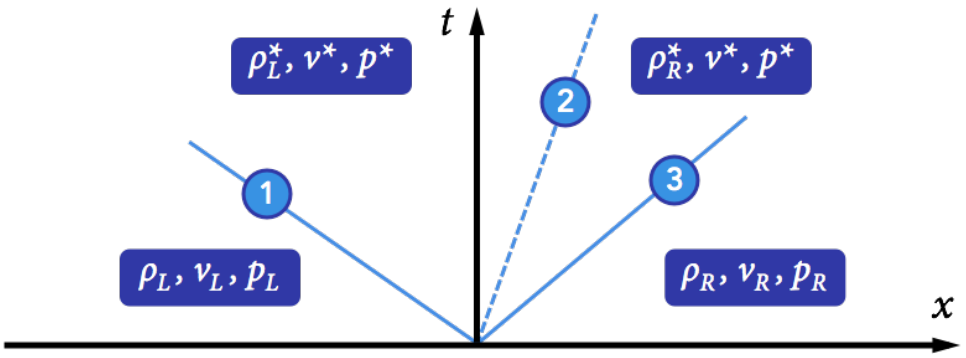


Figure 2.3: This figure depicts a typical solution of the Riemann problem for the Euler equations. There are three characteristic waves, travelling at velocities $v - c$, v and $v + c$ respectively, where v is the fluid velocity and c is the sound speed. The 1- and 3-waves can be either a shock wave or a rarefaction wave (with jumps in ρ, v and p). The 2-wave is a contact discontinuity (with a jump in ρ only).

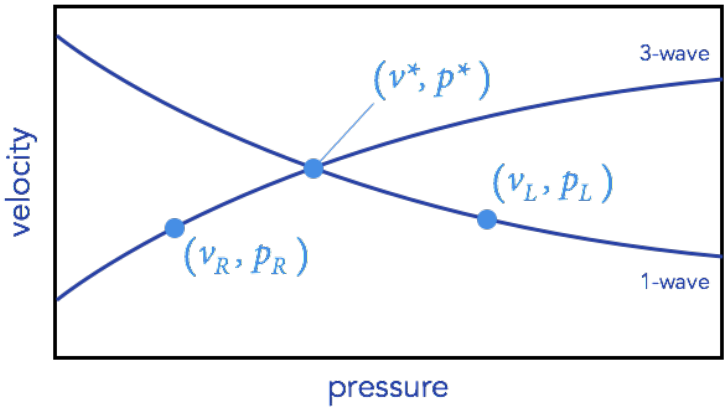


Figure 2.4: This diagram shows a typical Riemann problem, where the state space has been projected onto the pressure-velocity plane. The 1-wave through the left state and the 3-wave through the right state cross at only one point, the the intermediate state $(p*, v*)$.

2.4.4 Improving the Solve Stage

We now give a simplified picture of how exact Riemann solvers work, by considering the *state space* of all possible points (ρ, v, p). Our left and right input states are two points in this space: (ρ_L, v_L, p_L) and (ρ_R, v_R, p_R). From Fig. 2.3, we see that we are looking for:

- A state (ρ_L^*, v^*, p^*) connected to the state (ρ_L, v_L, p_L) by a 1-wave;
- A state (ρ_R^*, v^*, p^*) connected to the state (ρ_R, v_R, p_R) by a 3-wave;

which are also connected to each other by a 2-wave. We can simplify this problem by noting that $v = v^*$ and $p = p^*$ for *both* of these intermediate states, because only ρ jumps across the 2-

wave. This means we can project the full 3D state space into the v-p plane, as depicted in Fig. 2.4, in order to determine the values of v^* and p^*. The 1-wave travelling through the left state and the 3-wave travelling through the right state are shown as dark blue lines; (v^*, p^*) must be the point where both of these waves cross. (If there is more than one possible solution, we must use so-called entropy and evolutionary conditions to eliminate those that are physically or mathematically implausible. These conditions are described further in Chapter 5.) We can then determine ρ_L^* and ρ_R^* from the full 3D state space (where these waves may no longer intersect), in order to obtain the complete solution. We omit details of the algorithm here, but this is usually done with an iterative approach (see e.g. LeVeque, 2002).

The final step is to determine the state at the cell interface, which depends on the three wavespeeds. In the example depicted in Fig. 2.3, the 1-wave travels to the left (because $v - c < 0$), while the 2-wave travels to the right (because $v > 0$). Hence (ρ_L^*, v^*, p^*) is the state at the cell interface, represented by the t-axis. We can then substitute these values of ρ, v and p into the flux vectors defined in eq. (2.24) to obtain the required intercell fluxes.

A disadvantage of exact Riemann solvers is that the iterative approach is computationally expensive, especially when we need to solve a Riemann problem at every cell interface. For most of the merger simulations presented in this thesis, this means over two million Riemann problems *per timestep*. This issue will be amplifed further when we start considering the equations of magnetohydrodynamics, which involve *seven* characteristic waves. Another disadvantage is that they are arguably inefficient: they determine the complete solution to the Riemann problem, even though we only require the intercell fluxes.

For these reasons, *approximate* Riemann solvers are a popular alternative, which sacrifice some accuracy in favour of algorithmic simplicity and cheaper computational cost. For example: the HLL family of approximate Riemann solvers – named for their developers, Harten *et al.* (1983) – use the left and right input states to approximate the wavespeeds, and then use these to approximate the intermediate states using analytic formulae. These will be discussed further in Chapter 5, as part of the work completed on magnetic fields.

2.4.5 Improving the Reconstruct Stage

A disadvantage of Godunov's scheme is the assumption that the continuous function we are attempting to reconstruct is constant over a cell of width Δx. This, in turn, means it is only first-order accurate in space. However, it is possible to create a higher-order scheme by using a higher-order interpolation; we replace U_L^{i*} and U_R^{i*} in eq. (2.12) with $U_{L,\text{eff}}^{i*}$ and $U_{R,\text{eff}}^{i*}$. We use the subscript 'eff' for effective here, to highlight the fact that a Riemann problem is defined as having piecewise *constant* initial data. This is technically not true for a higher-order interpolation, but we assume it is in the area immediately surrounding the cell interface.

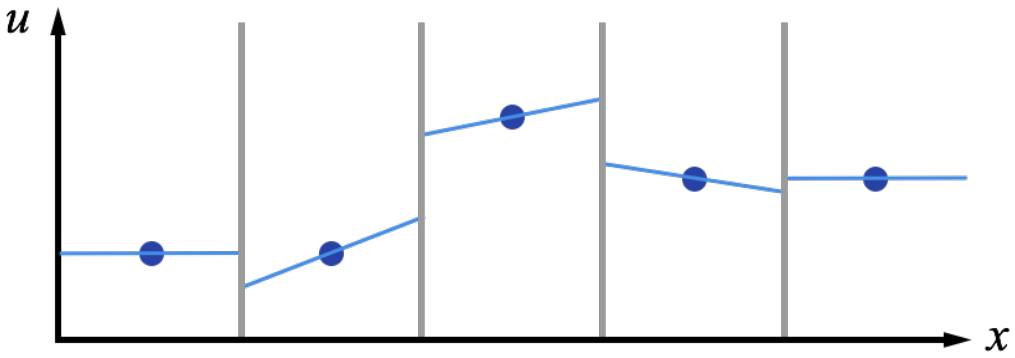


Figure 2.5: An example of a piecewise linear reconstruction, using the centred slope. It is clear that this reconstruction is not total variation diminishing, as the reconstructed function has introduced new extrema between the 1st and 2nd points, and between the 4th and 5th points.

Piecewise Linear Reconstruction and Total Variation

After piecewise constant, the next logical step is a piecewise *linear* reconstruction:

$$\tilde{u}_n(x, t_n) = U_n^i + (x - x_i)\sigma_i^n \qquad \text{for } x_{i*-1} \leq x \leq x_{i*}, \tag{2.39}$$

where σ_i^n is some slope. We can see that the cell average of this reconstruction will return U_n^i for each cell. There are a number of choices for the slope, such as the centred slope:

$$\sigma_{i,\text{centred}}^n = \frac{1}{2\Delta x}\left[U_{i+1}^n - U_{i-1}^n\right], \tag{2.40}$$

which is depicted in Fig. 2.5. However, this example shows a weakness of this approach: it can create new extrema around sharp discontinuities, which can lead to spurious oscillations. One measure of these oscillations is the 'Total Variation' (TV), and so we prefer a reconstruction scheme that is *Total Variation Dimishing* (TVD). This means that the total TV – and hence the amount of oscillations – does not increase with each step. It is possible to construct a piecewise-linear TVD scheme using *limiters*; examples of these can be found in chapter 6 of LeVeque (2002). We will come back to this reconstruction in Chapter 5.

Piecewise Parabolic Method

Charybdis currently uses the higher-order Piecewise Parabolic Method (PPM) of Colella and Woodward (1984). As the name suggests, this involves a piecewise-parabolic reconstruction:

$$\tilde{u}_n^i(x) = U_{-,i} + x(\Delta U_i + U_{6,i}(1 - x)), \tag{2.41}$$

where $U_{-,i} = \lim_{x \downarrow x_{i*-1}} \tilde{u}_n^i(x)$ and $U_{+,i} = \lim_{x \uparrow x_{i*}} \tilde{u}_n^i(x)$. The two coefficients are:

$$\Delta U_i = U_{+,i} - U_{-,i} \qquad U_{6,i} = 6(U_i^n - \tfrac{1}{2}(U_{-,i} + U_{+,i})). \tag{2.42}$$

This interpolation relies on determining appropriate values for $U_{-,i}$ and $U_{+,i}$, and the algorithm for doing so is given in section 1 of the original paper. (Note that this describes the method for a generic advection equation. Further modifications are required for the Euler equations, which are given in their section 3.) This scheme then enforces the TVD property with *monotonicity constraints*, which prevent the interpolation function from creating new extrema. We will also come back to this reconstruction in Chapter 5.

2.5 Modelling Time and Space

We conclude this chapter with some remarks on timestep restrictions, and the issues involved with modelling phenomena on different length scales.

2.5.1 Timesteps

At each iteration, the system is evolved for timestep Δt. This timestep must be chosen carefully: the smaller it is, the more iterations that are required to model a given amount of 'simulation' time, and hence the more real-world computation time that is required. However, as mentioned previously, there are also strict upper bounds: (a) neighbouring Riemann problems must not have enough time to start interacting with each other, and (b) the timestep must not violate the Courant-Friedrich-Lewy (CFL) stability condition.

The latter is a necessary (but not sufficient) condition for stability, and states that the *domain of dependence* of a numerical method, $\mathcal{D}_{\text{num}}(x,t)$, must include the domain of dependence of the PDE it is attempting to model, $\mathcal{D}_{\text{PDE}}(x,t)$ (LeVeque, 1992). $\mathcal{D}_{\text{PDE}}(x,t)$ is simply defined as the interval of points in the initial condition that influences the solution at (x,t). Since the Euler equations are hyperbolic, this interval is *bounded*: we can find it by simply tracing back along the characteristic waves from (x,t). Hence, if we alter the initial data *outside* this interval, we would not affect the solution at (x,t). Similarly, $\mathcal{D}_{\text{num}}(x = x_i, t = t_n)$ is the set of points in the initial data that could possibly influence the *numerical* solution at (x_i, t_n). So it is clear that we want the latter to include the former: we want to ensure that all points that *should* influence (x_i, t_n) are able to do so. However – as shown in Fig. 2.6 – increasing the timestep decreases the interval $\mathcal{D}_{\text{num}}(x,t)$. Hence, there must be some maximum timestep after which $D_{\text{PDE}} \nsubseteq D_{\text{num}}$, and it is this idea which gives us the CFL condition:

$$\Delta t \leq C\frac{\Delta x}{u}, \tag{2.43}$$

where Δx is the cell width, u is the maximum wavespeed and $C \in (0,1]$ is a parameter.

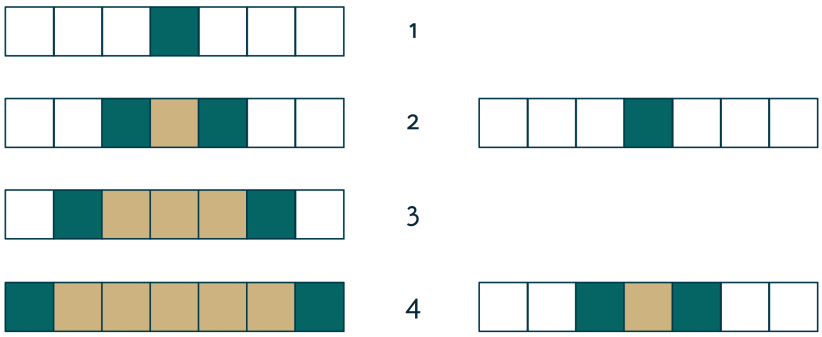


Figure 2.6: *The CFL Condition*: let's say we wish to model four units of time, using a 1D scheme where the value of a point at the next timestep depends on itself and its two neighbouring points at this timestep. If the timestep is one unit of time, then after four units of time, $\mathcal{D}_{\text{num}}$ is clearly 7 cells, as shown in the left-hand diagram. However, if the time step is two units of time, then $\mathcal{D}_{\text{num}}$ reduces to 3 cells, as shown on the right. This is the idea behind the CFL condition, as described in the main text.

2.5.2 Nested Grids

When creating the discretised grid, there is an issue we have not yet covered: the problem of modelling phenomena on different length scales. For example, we will see that a BH-NS merger can produce a long tidal tail, which can reach over 500 km in radius. To model this, we therefore require a computational grid at least 1,000 km square in the orbital plane. However, we are also interested in the detail on the surface of the neutron star, which is only $\sim 15 - 20$ km in radius. If we were to use a stiff *equation of state* (see Chapter 4), we would get a steep atmosphere, and we would therefore need several cells to accurately model the sharp drop in density from the neutron star core to empty space. But if we were to use a cell-width of, say, 1km, then this means at least *1,000 cells* in the x- and y-dirsections. We are also bound by the aforementioned CFL condition, which means that the whole grid - including uninteresting, empty regions - would have to be evolved by the same small timestep, since $\Delta t \propto \Delta x$.

Naturally, the large number of cells and the limited timestep will result in increased computational time. Thankfully, there is an elegant solution to this problem: to use *nested and refined grids*, where finer grids are only used over areas of interest (e.g. around the neutron star). For the results presented in this thesis, we use four levels of nested grids. Each grid uses 128 equidistant, Cartesian cells in the x- and y-direction, and 32 cells in the z-direction (where we assume the system is symmetric about the orbital plane, and so only half of the z-axis is simulated). Each grid is half the size of the next coarsest grid; the topmost grid covers a distance of 1,600km in the x-direction, and so the smallest cells have length 1.56 km. More details about how this is implemented can be found in Ruffert (1992).

Chapter 3
Further Numerical Physics

Black hole-neutron star mergers involve many more physical effects than just the flow of fluids. These range from the emission of neutrinos due to interacting particles, all the way to the emission of gravitational waves due to the relativistic warping of spacetime. In this chapter, we briefly describe the equations used to model these effects, as well as the numerical methods used to solve them. Note that these methods were not implemented as part of this thesis.

3.1 Physical Equations

3.1.1 The Modified Euler Equations

The additional physics modelled by Charybdis affect the three components of the linear momentum and the total energy. Hence, these effects are included as *source terms* on the right-hand side of the Euler equations (Ruffert *et al.*, 1996):

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho v^j}{\partial x^j} = 0, \tag{3.1}$$

$$\frac{\partial w^i}{\partial t} + \frac{\partial (\rho w^i v^j + p\delta^{ij})}{\partial x^j} = -\rho \frac{\partial \psi}{\partial x^i} - \rho \frac{\partial \phi}{\partial x^i} + V^j_{\text{mom}}, \tag{3.2}$$

$$\frac{\partial E}{\partial t} + \frac{\partial (E+p) v^j}{\partial x^j} = -\rho v^j \frac{\partial \psi}{\partial x^j} + W + S_E + V_{\text{E}}. \tag{3.3}$$

Note that we have switched to Einstein notation, where repeated indices imply a summation, $i, j \in \{1, 2, 3\}$ and $\{x^1, x^2, x^3\} = \{x, y, z\}$. Additionally, the velocity been split into the *kinematic* velocity $\mathbf{v}$ and the *dynamic* velocity $\mathbf{w}$. The dynamic velocity is the 'traditional' velocity, while the kinematic velocity is the relativistic momentum per unit rest mass (and hence reduces to the dynamic velocity in the Newtonian limit). The relation between them is given by eq. (3.5). This distinction is required to model the general-relativistic effect of gravitational waves in a Newtonian setting, using the method of Blanchet *et al.* (1990) (Section 3.2.4.)

Other terms: ψ is the Newtonian gravitational potential, ϕ is the backreaction potential due to gravitational waves, W is the energy source term due to gravitational waves, S_E is

the energy loss due to neutrinos, while V^i_{mom} and V_{E} are change in angular momentum and energy due to shear viscosity. The quantities E, $\mathbf{v}$, ϕ and $\ddot{D}_{ij}$ are defined by:

$$E = \rho\epsilon + \rho\frac{1}{2}w^i w^i, \tag{3.4}$$

$$v^i = w^i + \frac{4}{5}\frac{G}{c^5}\,\dddot{D}_{ij}\,w^j, \tag{3.5}$$

$$\phi = \frac{2}{5}\frac{G}{c^5}\left(R - \dddot{D}_{ij}\,x^j\,\frac{\partial\psi}{\partial x^i}\right), \tag{3.6}$$

$$\ddot{D}_{ij} = \text{STF}\left[2\int dV\left(2P\frac{\partial v_i}{\partial x^j} + \frac{\partial\psi}{\partial x^j}\left(x_i\,\frac{\partial\rho v^k}{\partial x^k} - 2\rho v_i\right) - \rho x_i\,\frac{\partial\dot{\psi}}{\partial x^j}\right)\right]. \tag{3.7}$$

The kinetic energy contribution to the total energy E is now calculated using the dynamic velocity w. Other terms: ϵ is the specific internal energy, G is the gravitational constant, c is the speed of light and D_{ij} is the quadrupole moment. Note that only the third time derivative of D_{ij} is required, so it is calculated directly using eq. (3.7). This involves the matrix operator STF – which stands for 'symmetric and trace free' – and is defined as:

$$\text{STF}[X_{ij}] = \tfrac{1}{2}X_{ij} + \tfrac{1}{2}X_{ji} - \tfrac{1}{3}\delta_{ij}X_{kk}. \tag{3.8}$$

Finally, ψ, $\dot{\psi}$ (the time derivative) and R are defined by Poisson equations:

$$\nabla^2\psi = 4\pi G\rho, \tag{3.9}$$

$$\nabla^2\dot{\psi} = -4\pi G\,\frac{\partial\rho v^i}{\partial x^i}, \tag{3.10}$$

$$\nabla^2 R = 4\pi G\,\dddot{D}_{ij}\,x^j\,\frac{\partial\rho}{\partial x^i}. \tag{3.11}$$

Note that the partial spatial derivative in eq. (3.10) comes from eq. (3.1).

3.1.2 Advection Equations

As well as the Euler equations, Charybdis also solves two additional advection equations for the electron number density n_e, and the entropy s:

$$\frac{\partial n_e}{\partial t} + \frac{\partial n_e v^j}{\partial x^j} = S_L, \tag{3.12}$$

$$\frac{\partial s n_b}{\partial t} + \frac{\partial (s n_b v^j)}{\partial x^j} = S_\nu + S_{\text{sh}} + S_{\text{visc}}. \tag{3.13}$$

n_e is required for the neutrino treatment, while s provides a useful way of determining the temperature of the system (Ruffert and Janka, 2001). Other terms: n_b is the baryon number density, and is given by $n_b = \rho/u_A$, where u_A is the atomic mass unit. n_e is then given by

$n_e = Y_e\, n_b$, where Y_e is the electron fraction. The matter is assumed to have no net charge, and so this is equal to the proton fraction, Y_p. S_L and S_ν are the change in lepton number and entropy due to neutrino emission, $S_{\rm visc}$ is the change in entropy due to shear viscosity, and $S_{\rm sh}$ is the change in entropy due to shock dissipation (see Ruffert and Janka, 2001).

Why use the entropy?

In our hydrodynamics code, the energy equation defined by eq. (3.3) is solved for the total specific energy, E, which is the sum of the specific internal energy, $E_{\rm int}$, and the kinetic energy, $E_{\rm int}$. This allows us to write the energy equation in a conservative form, and hence use the RSA strategy described in the previous chapter. However, a disadvantage of this approach is that the internal energy – which is required to determine the temperature – has to be calculated as the difference $E_{\rm int} = E - E_{\rm kin}$. This does not usually cause any problems, but in cases where $E_{\rm kin} \gg E_{\rm int}$ – such as in extremely degenerate neutron star matter – a small value of the internal energy can be signficantly affected by numerical noise.

Ruffert and Janka (2001) attempt to avoid this problem by using the *entropy* to calculate the temperature. However, the entropy depends on the values of the source terms in eq. (3.13), which in turn depend on the temperature. This necessitates the use of a *predictor-corrector* step, which ensures second-order accuracy for the time integration:

1. The source terms in eq. (3.13) are calculated using the old value of the temperature.
2. Eq. (3.13) then gives a new value for the entropy, and hence a new temperature.
3. Finally, these two steps are repeated, using an average value of the old and new temperatures. This gives a final value for the entropy.

This method is not fully consistent, and so can only be considered as an alternative approach rather than an inherently 'better' one. However, we can still use the internal energy to determine the temperature when the total and kinetic energies are not too large.

3.2 Numerical Methods

We now summarise the numerical methods used by Charybdis to determine the new source terms, and hence include the additional physical effects.

3.2.1 Modelling Neutrino Emission

The high temperatures and densities of BH-NS mergers mean that the remnant is mostly opaque to photons. The only possible cooling mechanism is via the emission of neutrinos and anti-neutrinos, which transport lepton number and energy away from the system (e.g. Rosswog, 2010). This emission is especially of interest in the context of GRBs, because some fraction will undergo $\nu\bar{\nu}$-annhilation, creating electron-positron pairs. These can in turn annihilate to produce photons, and hence feasibly accelerate a GRB-like relativistic jet. Charybdis approximates this emission with the neutrino leakage scheme of Ruffert *et al.* (1996).

At low optical depths, the equilibration of neutrinos with the stellar matter is the dominant effect. This means electron neutrinos are produced via *charged current β-processes*:

$$e^- + p \rightarrow n + \nu_e, \tag{3.14}$$

$$e^+ + n \rightarrow p + \bar{\nu}_e, \tag{3.15}$$

while *all* types of neutrinos (denoted ν_i, and including the heavier ν_μ, $\bar{\nu}_\mu$, ν_τ and $\bar{\nu}_\tau$ neutrinos) are created via *electron-positron pair annihilation* and by *plasmon decays* (where the latter process is dominant at high densities and electron degeneracies):

$$e^- + e^+ \rightarrow \nu_i + \bar{\nu}_i, \tag{3.16}$$

$$\bar{\gamma} \rightarrow \nu_i + \bar{\nu}_i. \tag{3.17}$$

The neutrino emission – and hence the loss of energy and lepton number – can be computed directly from the rates of these processes. At higher optical depths, the timescale for equilibration is much shorter, so neutrinos are simply assumed to be at their equilibrium abundances. Instead, *neutrino diffusion* becomes the dominant process.

At intermediate optical depths, this loss is a superposition of these two regimes; the 'weighting' between the two is a function of the neutrino opacity. The neutrino opacity is determined by similarly considering the rates of relevant processes. For electron neutrinos, the opacity is dominated by the inverse of the β-processes above. For heavier neutrinos, it is dominated by neutral-current charge scattering off nucleons:

$$\nu_i + \left\{ \begin{array}{c} n \\ p \end{array} \right\} \rightarrow \nu_i + \left\{ \begin{array}{c} n \\ p \end{array} \right\} \tag{3.18}$$

All the values that are required – the neutrino opacity, the neutrino emission rates and the loss of energy and lepton number – can be determined by analytic formulae, which are given in the appendices of Ruffert *et al.* (1996). These are then summed over all cells on the grid to find the total loss in energy and lepton number, and hence the source terms S_E and S_L required for eqs. (3.3) and (3.12). Finally, the change in entropy due to neutrinos – and hence the source term S_ν required for eq. (3.13) – is given by (Ruffert and Janka, 2001):

$$S_\nu = \frac{1}{k_B T} \left[S_E - S_L \left(\mu_e + \mu_p - \mu_n \right) \right], \tag{3.19}$$

where the μ terms are the chemical potentials of electrons, protons and neutrons. These are given by the realistic equation of state table; this will be described further in Chapter 4.

3.2.2 Modelling Viscosity

In an accretion disk, the radial flux of angular momentum has two opposing components: an inward flux, due to accreting matter in nearly circular orbits, and an outward flux, due to

turbulence and magnetic processes. The latter is often approximated as an advanced viscous flux (Balbus and Hawley, 1998).[1] However, recall that the Euler equations do not include any viscosity terms (unlike the more general and physically accurate Navier-Stokes equations). Hence, we approximate the effects of viscosity using an α-model, where α is the disk-viscosity parameter introduced by Shakura and Sunyaev (1973). This involves calculating the following source terms, which are added to eqs. (3.2), (3.3) and (3.13):

$$V^i_{\text{mom}} = \frac{\partial}{\partial x^j}\left[\eta\left(\frac{\partial v^i}{\partial x^j} + \frac{\partial v^j}{\partial x^i} - \frac{2}{3}\delta_{ij}\frac{\partial v^k}{\partial x^k}\right)\right], \tag{3.20}$$

$$V_E = \frac{\eta}{2}\left(\frac{\partial v^i}{\partial x^j} + \frac{\partial v^j}{\partial x^i} - \frac{2}{3}\delta_{ij}\frac{\partial v^k}{\partial x^k}\right)^2, \tag{3.21}$$

$$S_{\text{visc}} = \frac{1}{k_B T}\left[\frac{\eta_p}{2}\left(\frac{\partial v^i}{\partial x^j} + \frac{\partial v^j}{\partial x^i} - \frac{2}{3}\delta_{ij}\frac{\partial v^k}{\partial x^k}\right)^2\right] \tag{3.22}$$

All of the derivatives are approximated using finite differences. The parameter η is given by:

$$\eta = \alpha\rho c_s^2/\Omega_K \tag{3.23}$$

where Ω_K is the Keplerian angular velocity and c_s is the adiabatic sound speed. We set $\alpha = 0.1$, following the work of Setiawan *et al.* (2006). In comparison, *numerical* viscosity corresponds to a value of approximately $\alpha = 0.01$ (e.g. Ruffert and Janka, 2001). Further details of the implementation can be found in Setiawan *et al.* (2006).

3.2.3 Modelling Self-Gravity

Self-gravity is the gravitational force exerted on a body due to its own mass, and is therefore important in the context of BH-NS mergers. It is encapsulated by the Newtonian gravitational potential ψ, and so to model its effects, this quantity must be determined everywhere on the grid. We start with Gauss's law for gravity, which defines the gravitational field $\mathbf{g}$:

$$\nabla \cdot \mathbf{g} = -4\pi G\rho. \tag{3.24}$$

We know that the gravitational field is conservative, so we can write $\mathbf{g}$ in terms of the Newtonian gravitational potential: $\mathbf{g} = -\nabla\psi$. This gives us the Poisson equation in eq. (3.9). Using a Green's function, we can solve this equation to obtain:

$$\psi(\mathbf{r}) = -\int \frac{\rho(\mathbf{r}')\, d^3\mathbf{r}'}{|\mathbf{r} - \mathbf{r}'|}. \tag{3.25}$$

[1] Balbus and Hawley (1998) note that the turbulence is not generally reducible in this way, but it is a plausible approximation for 'gross physical properties' such as heating.

This integral is treated as a *convolution* between $\rho(\mathbf{r})$ and the position function $\mathbf{r}$, and so the solution can be determined using Fourier transforms:

$$\psi = \mathcal{F}^{-1}[\mathcal{F}(\mathbf{r})\mathcal{F}(\rho)]. \tag{3.26}$$

Charybdis uses Fast Fourier Transform algorithms. The spatial derivatives of ψ are then determined using finite differences, and added as source terms to eq. (3.2) and (3.3). A similar approach is used to determine the time-derivative $\dot{\psi}$ and the quantity R, which are given in eqs. (3.10) and (3.11). This procedure is slightly complicated by the use of nested grids; further details of the implementation can be found in Ruffert (1992).

3.2.4 Modelling Gravitational Waves

Gravitational waves are ripples in spacetime, and a consequence of general relativity. They are emitted from accelerating masses, in a similar manner to how electromagnetic waves are emitted from accelerating charges. However, the weakness of the gravitational interaction means they have a much smaller magnitude: despite being predicted by Einstein in 1919, they are yet to be detected directly.[2] BH-NS mergers are therefore of interest to researchers in this area: the large masses involved means that they are promising sources of gravitational waves that could be detected from Earth. Additionally, it is hard to localise gravitational wave signals with current technological limits. However, if the merger results in a GRB, it *is* possible to localise the corresponding electromagnetic signal.

Charybdis cannot make any accurate qualitative predictions about the gravitational wave signal, as it is fundamentally Newtonian. However, their effect must still be approximated and included in the simulation, because it is the emission of these waves that carry angular momentum away from the system, causing the two objects to in-spiral and coalesce. The gain in energy due to gravitational waves is included as a source term W in eq. (3.3):

$$W = -\rho v^j \frac{\partial \phi}{\partial x^j} + \frac{4}{5}\frac{G}{c^5}\,\dddot{D}_{ij}\,v^i\left(\rho\frac{\partial \psi}{\partial x^j} + \frac{\partial P}{\partial x^j}\right). \tag{3.27}$$

Again, finite differences are used to evaluate the derivatives required. The total emitted luminosity of gravitational waves, $\mathcal{L}$, can be estimated by summing W over the emitting volume, or by using the classical quadrupole formula (Ruffert *et al.*, 1996):

$$\mathcal{L} = \frac{1}{5}\frac{G}{c^5}\,\dddot{D}_{ij}\,\dddot{D}_{ij}. \tag{3.28}$$

Both these equations require the term $\dddot{D}_{ij}$, the third time derivative of the quadrupole moment, given by (3.7). This dependence characterises the fact that gravitational waves are only emitted from a rotating mass system when it is asymmetrical about its axis of rotation. If the system is symmetric, then $D_{ij} = 0$, and $\mathcal{L}$ and W also reduce to zero.

[2] However, the first indirect detection of primordial gravitational waves has very recently been announced by the BICEP2 collaboration. For more information, see: `http://bicepkeck.org/`.

3.3 Black Holes and Neutron Stars

Finally, we conclude this chapter with a summary of the black hole and neutron star models.

3.3.1 Modelling the Black Hole

The black hole starts as a point mass. However, this alone is not sufficient; CHARYBDIS is fundamentally Newtonian, and so important general-relativistic effects will not arise naturally:

- the existence of the innermost stable circular orbit (ISCO);
- the existence of the event horizon;
- the lack of angular momentum barrier.

As with the treatment of gravitational waves, these effects have to be explicitly approximated.

The Innermost Stable Circular Orbit

As the name suggests, the *innermost stable circular orbit* is the smallest radius $r = r_{\mathrm{ISCO}}$ at which a particle's orbit is stable to minor perturbations. At $r < r_{\mathrm{ISCO}}$, such a perturbation could send the particle either through the event horizon (and into the black hole) or out to infinity. It is physically important, as it usually defines the inner edge of an accretion disk. It also affects the ultimate fate of the merger: the neutron star needs to be tidally disrupted *before* it reaches the ISCO, otherwise it will be absorbed by the black hole before it can form a torus sufficiently massive to power a GRB (Miller, 2005; Ruffert and Janka, 2010).

It is possible to model the ISCO in a Newtonian setting with a GR-mimicking potential, such as that proposed by Paczyński and Wiita (1980) for a non-rotating black hole. CHARYBDIS uses the more recent potential of Artemova *et al.* (1996), which extends the Paczyński-Wiita potential by including a dependence on the black hole spin. This is done using the *Kerr parameter*, a. This gravitational potential, which we call ψ_{BH}, is given by:

$$\frac{d\psi_{\mathrm{BH}}}{dr} = -\frac{GM_{\mathrm{BH}}}{r^{2-\beta}(r - r_{\mathrm{H}})^{\beta}}, \tag{3.29}$$

where $r_H(a)$ is the event horizon of the black hole, and β is given by:

$$\beta = \frac{r_{\mathrm{ISCO}}(a) - r_{\mathrm{H}}(a)}{r_{\mathrm{H}}(a)} \tag{3.30}$$

Details of the implementation of this potential are given in Ruffert and Janka (2010).

The Event Horizon and the Angular Momentum Barrier

Perhaps the most well-known GR effect is the existence of the event horizon around the black hole: an area within which the escape velocity is sufficiently large to stop light (and hence anything else) from escaping. This is simply modelled as a *vacuum sphere* around the point mass: any matter that enters this sphere is removed from the grid, and its mass, momentum and angular momentum are added to the black hole. The position of the black hole is then

shifted accordingly. The radius of the vacuum sphere is chosen as the arithmetic mean of the event horizon and the ISCO (Ruffert and Janka, 2010).

A relativistic BH also lacks an *angular momentum barrier*: it is possible for a particle with non-zero angular momentum to reach $r = 0$ (if it crosses the event horizon). This is not possible in purely Newtonian physics: a particle with a given angular momentum has only one possible orbit. This effect is also accounted for by the vacuum sphere.

3.3.2 Modelling the Neutron Star

The neutron star is modelled as a sphere of high-density fluid in hydrostatic equilibrium: the thermal pressure of the neutron star is balanced by its self-gravity. The model is spherically symmetric, and so the problem of constructing it is reduced to finding four one-dimensional radial profiles: the 'contained mass' $m(r)$ (the mass contained within volume $\frac{4}{3}\pi r^3$), the pressure $p(r)$, the mass density $\rho(r)$ and the proton fraction $Y_p(r)$. We obtain these profiles by solving the Newtonian equations of hydrostatic equilibrium:

$$\frac{dm(r)}{dr} = 4\pi r^2 \rho(r) \tag{3.31}$$

$$\frac{dp(r)}{dr} = \frac{Gm(r)\rho(r)}{r^2} \tag{3.32}$$

where G is the gravitational constant. The $m(r)$ and $p(r)$ profiles are found by solving eqs. (3.31) and (3.32) with a fourth-order Runge-Kutta scheme. Simultaneously, $Y_p(r)$ is found using the approach of Oechslin *et al.* (2007), which assumes that the NS is in neutrinoless beta-equilibrium ($\mu_\nu = \mu_n - \mu_p - \mu_e = 0$) with temperature $k_B T = 0.1$ MeV. Finally, $\rho(r)$ is found using the equation of state (EOS) table. Because of this dependence on the EOS, a new NS must be constructed for each EOS that we use; this is discussed further in Chapter 4.

Part II

Investigations

Chapter 4
The Equation of State

In this chapter, we aim to investigate the effect of the equation of state (EOS) on black hole-neutron star mergers, focusing on the dynamics of the merger, the properties of the resultant disk and the neutrino emission. We present results of numerical simulations using three realistic, tabular EOSs, developed by H. Shen *et al.* (1998a b), G. Shen *et al.* (2011b) and G. Shen *et al.* (2011a) respectively.

Note that – at the time of writing – this work is being prepared for publication in the *Monthly Notices of the Royal Astronomical Society*.

4.1 Introduction

4.1.1 Motivation

As we saw in Chapter 2, the Euler equations of hydrodynamics involve six unknown quantities (ρ, ρu_x, ρu_y, ρu_z, $\mathcal{E}$ and p)[1], but only five equations. Hence, we require another equation to close the system before we can find unique solutions. This is the *equation of state* (EOS), and encapsulates the relationship between the pressure of a fluid and its other properties, such as its mass density, temperature and composition. Although the neutron star EOS is well understood at low densities, it is yet to be constrained at supranuclear densities (that is, above the nuclear saturation density, $\rho \sim 10^{14.7}$ g cm^{-3}). Consequently, the EOS must be treated as a parameter in numerical studies of black hole-neutron star (BH-NS) mergers, alongside e.g. the mass ratio, the initial separation and the black hole spin.

This choice of EOS can influence the dynamics and outcome of the merger, but the extent of this influence is still to be fully determined. For example, a 'stiffer' EOS (which returns a higher pressure at supranuclear densities) will produce a neutron star with larger radius than one of the same mass produced by a 'softer' EOS. This will be tidally disrupted by the black hole at a larger orbital separation, feasibly resulting in a larger disk mass (Shibata and Taniguchi, 2011). Additionally, the EOS influences shock evolution, which will affect the disk temperature (Shen *et al.*, 2011b). The disk mass and disk temperature both determine the

[1] That is: mass density, the three components of the linear momentum, total energy and pressure.

amount of energy that can be released through neutrino processes, and hence the properties of the resulting neutrino-driven GRB (or indeed, whether one is produced at all). It is therefore important to investigate the effect of the EOS on BH-NS mergers.

4.1.2 Types of EOS

Comprehensive reviews of the types of EOS used in previous numerical studies of compact object mergers can be found in e.g. Shibata and Taniguchi (2011, BH-NS) and Faber and Rasio (2012, NS-NS). We provide a brief summary here.

The Polytropic and Piecewise-Polytropic EOS

A popular choice for general-relativistic simulations (but also investigated by pseudo Newtonian codes, e.g. Ruffert and Janka, 2010) is the polytropic EOS. Here, the pressure p, the density ρ and the specific internal energy ϵ are related by the equation:

$$p = (\Gamma - 1)\,\rho\epsilon = \kappa\rho^{\Gamma}, \tag{4.1}$$

where κ is the *polytropic constant*, and Γ is the *adiabatic index* (with $\Gamma = 2$ being a common choice). While this is simple to implement, an obvious disadvantage is that it is not realistic: it is known to disagree with nuclear theory (Shibata and Taniguchi, 2011).

An extension to this idea is the piecewise-polytropic EOS, where the EOS is divided into a number of conjoined 'segments', each with its own values for Γ and κ:

$$p = \kappa_i\,\rho^{\Gamma_i}, \qquad \rho_{i-1} \leq \rho < \rho_i, \qquad (1 \leq i \leq n), \tag{4.2}$$

where n is the number of segments, and ρ_i represents the 'boundaries' between each segment (with $\rho_0 = 0$ and $\rho_n \to \infty$). It has been shown that if you assume a fixed EOS at lower densities (which is well understood anyway), then it would be possible to replicate all known cold EOS models with just three additional segments, with boundaries at $\rho = 10^{14.7}$ g cm^{-3} (the nuclear saturation density) and $\rho = 10^{15}$ g cm $^{-3}$.

However, this EOS still has no temperature or composition dependence. While this is fine for the in-spiral and early parts of the merger, it becomes an issue at later stages. Once the NS becomes tidally disrupted, then heating effects (such as shock heating, viscous heating and neutrino cooling) and changes in the composition of the disk (due to neutrino emission) become significant. The composition dependence is especially important if we wish to investigate neutrino-antineutrino annihilation in the context of GRBs.

The Realistic EOS

A solution to these issues is to use a 'realistic' EOS, which has been derived from nuclear theory, and includes both temperature and composition dependence. These take the form:

$$p = p(\rho, k_B T, Y_p), \tag{4.3}$$

where k_B is Boltzmann's constant, and Y_p is the proton fraction. Such EOSs are usually too sophisticated to calculate at run-time, and so we use a pre-calculated EOS *table*. This is a three dimensional table, where the three parameters – ρ, $k_B T$ and Y_p – form the three axes. Given the value of these three co-ordinates, we can look up the corresponding value of several thermodynamic quantities, such as the pressure, the chemical potential or the entropy. A three-dimensional linear interpolation is used for values that are not explicitly listed.

As described above, the NS EOS has not been constrained at supranuclear densities. Hence, there are a number of competing realistic EOSs, which have been developed using different models and approximations. In this chapter, we present the first BH-NS simulations to use two such EOSs, developed by G. Shen *et al.* (2011b) and G. Shen *et al.* (2011a). For reference, we compare these with the previously implemented EOS of H. Shen *et al.* (1998a b).

4.1.3 Review of Realistic EOSs

Previous EOSs

The first widely-used realistic EOS for compact object merger simulations was that of Lattimer and Swesty (1991). This uses the compressible liquid drop model, based on the non-relativistic Skyrme-Hartree-Fock framework. It has been investigated in the context of both NS-NS mergers (e.g. Ruffert *et al.*, 1996) and BH-NS mergers (e.g. Deaton *et al.*, 2013). However, it was designed for use with supernova simulations, and so – as noted by Rosswog and Davies (2002) – it does not cover the complete range of parameters required for modelling mergers. Additionally, the non-relativistic approach cannot reproduce some key effects, such as the saturation property of nuclear matter (Shen *et al.*, 1998a).

These issues were addressed by the EOS of H. Shen *et al.* (1998a b) – henceforth referred to as **H98-TM1** – which uses the relativistic mean free (RMF) model combined with the TM1 parameter set to approximate the relativistic Dirac-Bruckner-Hartree-Fock framework. This EOS has also been investigated in the context of both NS-NS mergers (e.g. Rosswog and Davies, 2002) and BH-NS mergers (e.g. Archibald, 2008; Duez *et al.*, 2010).

Above $\rho \sim \rho_0/3$ (where ρ_0 is the nuclear saturation density), matter is treated as a simple homogeneous fluid. Below this density, it becomes an inhomogeneous mixture of electrons, muons, free nucleons and a single species of heavy nuclei, whose mass is given by a statistical average. This inhomogeneous matter is modelled using the Thomas-Fermi approximation: a lattice of spherical heavy nuclei, which may be surrounded by a nucleon gas and a lepton gas. (Note that the lepton contribution was not included in the original H98-TM1 table, and so must be added separately as a gas of uniform, non-interacting relativistic particles.) Finally, for extremely low densities, matter is treated as a uniform nucleon gas.

The G. Shen EOSs

In this chapter, we present the first BH-NS merger simulations to use the two newer EOSs of G. Shen *et al.* (2011b, NL3) and G. Shen *et al.* (2011a, FSU), henceforth referred to as **G11-NL3** and **G11-FSU** respectively. Both of these also rely on the RMF model for homogeneous matter at high densities and inhomogeneous matter at intermediate densities. However, G11-NL3

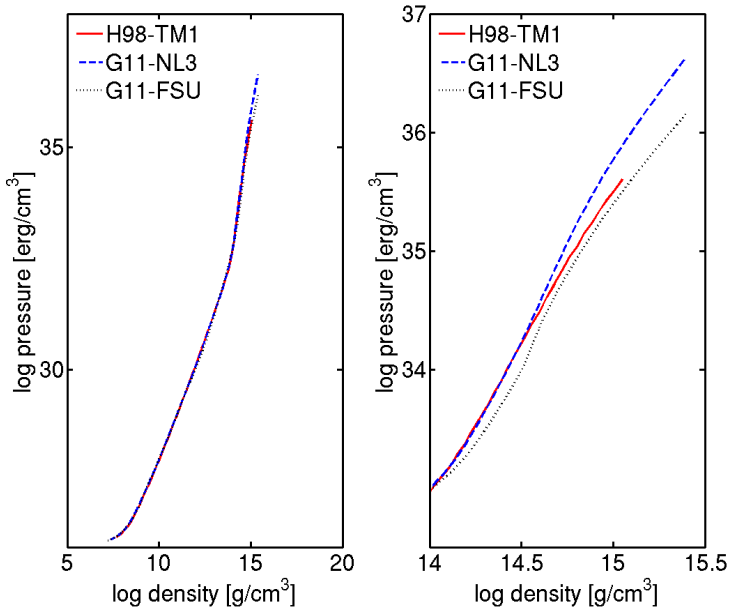


Figure 4.1: Pressure against density for the three EOSs, for Y_p = 0.1 and k_BT = 1 MeV. Both plots are of the same data, with the right plot zoomed in on the high-density regime. The (modified) NL3 parameter set is stiffer than TM1 and (modified) FSUGold.

	ρ (g cm^{-3})		Y_p		k_BT (MeV)	
EOS	**Range**	**Pts**	**Range**	**Pts**	**Range**	**Pts**
H98-TM1	$5.01 \times 10^7 - 1.12 \times 10^{15}$	180	0.015 - 0.56	50	0.1 - 97.72	120
G11-NL3	$1.66 \times 10^7 - 2.63 \times 10^{15}$	328	0, 0.05 - 0.56	53	0, 0.158 - 79.4	109
G11-FSU	$1.66 \times 10^7 - 2.48 \times 10^{15}$	328	0, 0.05 – 0.56	53	0, 0.158 - 74.9	109

Table 4.1: Ranges of values and number of points for the three EOSs investigated in this chapter: H98-TM1 (H. Shen *et al.*, 1998a b), G11-NL3 (G. Shen *et al.*, 2011b) and G11-FSU (G. Shen *et al.*, 2011a).

uses a modified version of the NL3 parameter set[2], which is stiffer than TM1 at high densities, while G11-FSU uses a modified version of the FSUGold parameter set[3], which has a similar stiffness to TM1 at high densities. These differences are shown in Fig. 4.1. Additionally, the two G11 EOSs cover different ranges in density, temperature and proton fraction to H98-TM1. These ranges – along with the number of tabulated points – are given in Table 4.1.

As well as the different parameter sets, both of these EOSs use *exact solutions* to the RMF

[2] The modification to NL3 was to add a density-dependent coupling, in order to create a more physical model of neutron matter at very low density. Details of this modification are given in G. Shen *et al.* (2010).

[3] The modification to FSUGold was made so that it could support a larger maximum neutron star mass (2.1 $M_\odot$ instead of the original 1.7 $M_\odot$), in order to accommodate observational evidence of neutron stars with masses greater than 1.7 $M_\odot$. Details of this modification are given in G. Shen *et al.* (2011a).

Quantity	ρ	$k_B T$	Y_p	p	$\rho\epsilon$	Γ_c	$\mu_{n,p}$	s/A
Charybdis	g cm^{-3}	MeV	-	MeV cm^{-3}	MeV cm^{-3}	-	MeV	k_B

Quantity	n_B	$k_B T$	Y_p	p	F/A	Γ_c	$\mu_{n,p}$	s/A
G11 EOSs	fm^{-3}	MeV	-	MeV fm^{-3}	MeV	n/a	MeV	k_B

Table 4.2: The top table shows the quantities expected by Charybdis, while the bottom table shows the equivalent quantities given in the original versions of the two G11 EOSs. Differences are highlighted by red text. Γ_c is the adiabatic index, $\mu_{n,p}$ are the neutron/proton chemical potentials and s/A is the entropy. Note that 'n/a' signifies that the quantity is not available.

equations – rather than the Thomas-Fermi and variational approximations used by H98-TM1 – which means they can model the shell structure of nuclei. Furthermore, at low densities, they both use a virial expansion for a non-ideal gas consisting of free nucleons, alpha particles and *thousands* of species of heavy nuclei. This differs from H98-TM1, which used only a single averaged species, and results in differences in the composition of nuclear matter. It is plausible that the emission of neutrinos – and hence the properties of any neutrino-driven GRB – will be sensitive to these shell effects and differences in composition.

4.2 Implementing the G11 EOSs

The H98-TM1 has been previously implemented in Charybdis (Archibald, 2008). In this section, we describe the two stages required to implement the two G11 EOSs: (a) adjusting the quantities and their units so that they are compatible with what is expected by Charybdis, and (b) calculating the adiabatic index, which was not included in the tables.

4.2.1 Stage One: Adjusting Quantities and Units

Table 4.2 lists the quantities and units expected by Charybdis, and those used by the two G11 EOSs. The differences are highlighted in red, and must be adjusted accordingly. The simplest changes are the mass density and pressure, which are adjusted as follows:

$$\rho = n_B \, m_u \times 10^{39}, \tag{4.4}$$

$$\tilde{p} = p \times 10^{39}, \tag{4.5}$$

where m_u is the atomic mass unit, and $\tilde{p}$ is the required form of the pressure. The factor of 10^{39} converts both quantities from units of fm^{-3} to units of cm^{-3}.

Less trivial is the change in energy. Charybdis expects an internal energy density, whereas the G11 EOSs list a free energy per baryon. To convert between these, we rearrange the defi-

nition of the (Helmholtz) free energy to get the internal energy per baryon, $\mathcal{U}$:

$$\mathcal{U} = \mathcal{F} + Ts + M - m_u, \tag{4.6}$$

where $\mathcal{F}$ is the free energy per baryon, T is the temperature and s is the entropy. The $+M$ term is required because the free energy is defined relative to the free nucleon mass M; the $-m_u$ term is required because the internal energy needs to be defined relative to the atomic mass unit m_u. And so, to obtain the internal energy *density*, we multiply $\mathcal{U}$ by the number of baryons per unit volume, which is listed in the EOS table as the baryon number density, n_B:

$$\rho\epsilon = (\mathcal{F} + Ts + M - m_u)\, n_B. \tag{4.7}$$

4.2.2 Stage Two: Calculating the Adiabatic Index

We see in Table 4.2 that the two G11 EOSs do not include data for the adiabatic index, Γ_c, which describes the stiffness of the EOS at constant entropy. It is defined as:

$$\Gamma_c = \left(\frac{\partial \ln p}{\partial \ln \rho}\right)_s, \tag{4.8}$$

where s is the entropy. Before we can use the EOS, we need to calculate this value for every point in the three-dimensional table, $(\rho, k_B T, Y_p)$. For completeness, we summarise the procedure here. It can be divided into two parts: finding the direction of constant entropy, s, and then evaluating the derivative along this direction.

Finding the Direction of Constant Entropy

a. We start with a point $(\rho_i, k_B T_i, Y_p)$ where the index i refers to the density index within the table. We use the EOS table to look up the entropy at this point, s_i.

b. Given fixed values of Y_p and s_i, we use a bisection iteration to estimate the temperature at the points ρ_{i-2} and ρ_{i+2}. We call these temperatures $k_B T_{i-2}$ and $k_B T_{i+2}$. (In other words, given $\rho_{i\pm2}$ and Y_p, we estimate what values of $k_B T_{i\pm2}$ will return s_i.)

c. Our direction of constant entropy in the $k_B T - \rho$ plane is then given by the vector: $(k_B T_{i+2} - k_B T_{i-2}, \rho_{i+2} - \rho_{i-2})$.

d. Finally, we use these points to determine the intermediate temperatures $k_B T_{i-1}$ and $k_B T_{i+1}$. We use a linear interpolation to do this, rather than repeating the bisection iteration procedure, in order to reduce the noise in the direction of constant entropy.

Calculating the Derivative

e. Using these five values of $k_B T$ and ρ (and the fixed value of Y_p), we use the EOS table to look up the pressure values at the points of interest: p_{i-2}, p_{i-1}, p_i, p_{i+1} and p_{i+2}.

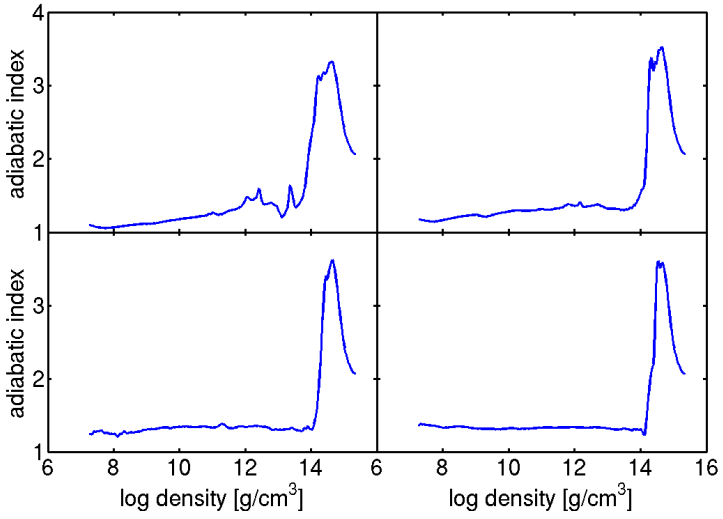


Figure 4.2: The adiabatic index, calculated for G11-NL3 using the method described in Section 4.2.2. Plots made at constant entropy s=1, for four values of the proton fraction: Y_p = 0.1 (top left); Y_p = 0.2 (top right); Y_p = 0.3 (bottom left); Y_p = 0.4 (bottom right).

f. Finally, we take logs of the pressure and density values, and then approximate the required derivative using finite differences. We use a fourth-order centred stencil:

$$\Gamma_c \approx \frac{1}{12\Delta \ln \rho} \left(\ln p_{i-2} - 8 \ln p_{i-1} + 8 \ln p_{i+1} - \ln p_{i+2} \right). \tag{4.9}$$

Some results from this calculation are shown in Fig. 4.2. The adiabatic index is given for entropy s = 1, and for four values of the proton fraction, Y_p = 0.1, 0.2, 0.3 and 0.4. These results compare favourably with the results presented in fig. 8 of Shen *et al.* (2011b). Given that they were computed independently, we can be assured that our method is reliable.

4.3 Initial Conditions

The initial conditions for our runs are given in Table 4.3. We present results from six runs: one for each EOS, for two mass ratios, $q = M_{\rm BH}/M_{\rm NS} = 3$ and $q = 5$. The initial separation is sufficiently large to allow a number of orbits before coalescence.

The black hole and neutron star models are constructed using the methods described in Section 3.3. The NS mass is fixed at $M_{\rm NS} = 1.6\ M_\odot$, while the spin is set to zero. Recall that we have to construct a new NS for each EOS that we use: the radial profiles of the density, the proton fraction and the pressure are given in Fig. 4.3, while their properties are given in Table 4.3. The effect of the EOS is clear: the softer G11-FSU NS results in a larger central density and pressure, and a smaller radius. We also note that the difference in radius for the HS98-TM1 and GS11-NL3 NSs is much smaller than the smallest length that can be resolved by our simulation (1.56 km on the finest grid), and so they are, in effect, equal.

Model	EOS	q	s	$R_{\rm NS}$ km	$\mathcal{C}$	ρ^c g cm^{-3}
H-TM1-3	H98-TM1	3	0.5	17.6	0.135	3.70×10^{14}
G-NL3-3	G11-NL3	3	0.5	17.8	0.133	3.55×10^{14}
G-FSU-3	G11-FSU	3	0.5	16.4	0.144	4.53×10^{14}
H-TM1-5	H98-TM1	5	0.5	17.6	0.135	3.70×10^{14}
G-NL3-5	G11-NL3	5	0.5	17.8	0.133	3.55×10^{14}
G-FSU-5	G11-FSU	5	0.5	16.4	0.144	4.53×10^{14}

Table 4.3: Initial conditions for all runs. $q = M_{\rm BH}/M_{\rm NS}$ is the mass ratio, s is the BH spin (taken into account by the GR-mimicking potential as described in Section 3.3.1), $R_{\rm NS}$ is the NS radius, $\mathcal{C} = GM_{\rm NS}/c^2R_{\rm NS}$ is the NS compaction, and ρ^c is the central density of the NS.

4.4 Results and Discussion

4.4.1 The Time Axis

For each set of three cases (corresponding to fixed q), the NS starts at the same separation from the BH at time $t = 0$. However, because of the different NS compactions, we expect events in the dynamical evolution of the NS to happen at different times. For example, the less compact G11-FSU NS will be tidally disrupted by the BH sooner. These differences in time are not of interest, as they are only a consequence of the fixed initial separation. Hence, we remove them from the results by shifting the time axis for the G11-NL3 and G11-FSU results so that the first major mass transfer episode – defined as the time of the peak accretion rate – happen at the same time as for the H98-TM1 results. This can be seen in Figs. 4.4 and 4.5, where the first peaks all occur at ~ 12 ms and ~ 7 ms for $q = 3$ and $q = 5$ respectively.

4.4.2 Inspiral and Merger

The accretion rate as a function of time for each EOS is given in Figs. 4.4 ($q = 3$) and 4.5 ($q = 5$), and is calculated as the time derivative of the BH mass. The values of the peak accretion rate are given in Table 4.5. The BH-NS separation is given in Figs. 4.6 ($q = 3$) and 4.7 ($q = 5$). To calculate this, the NS position is defined as the centre of mass of a small sphere centred on the density maximum (excluding the BH). Also note that this separation is only given for the lifetime of the NS, before it is completely tidally shredded.

Results

$q = 3$: in all three cases, the neutron star rapidly plunges until it reaches a separation of $\sim$ 60 km from the black hole, when the first mass transfer episode occurs. This corresponds

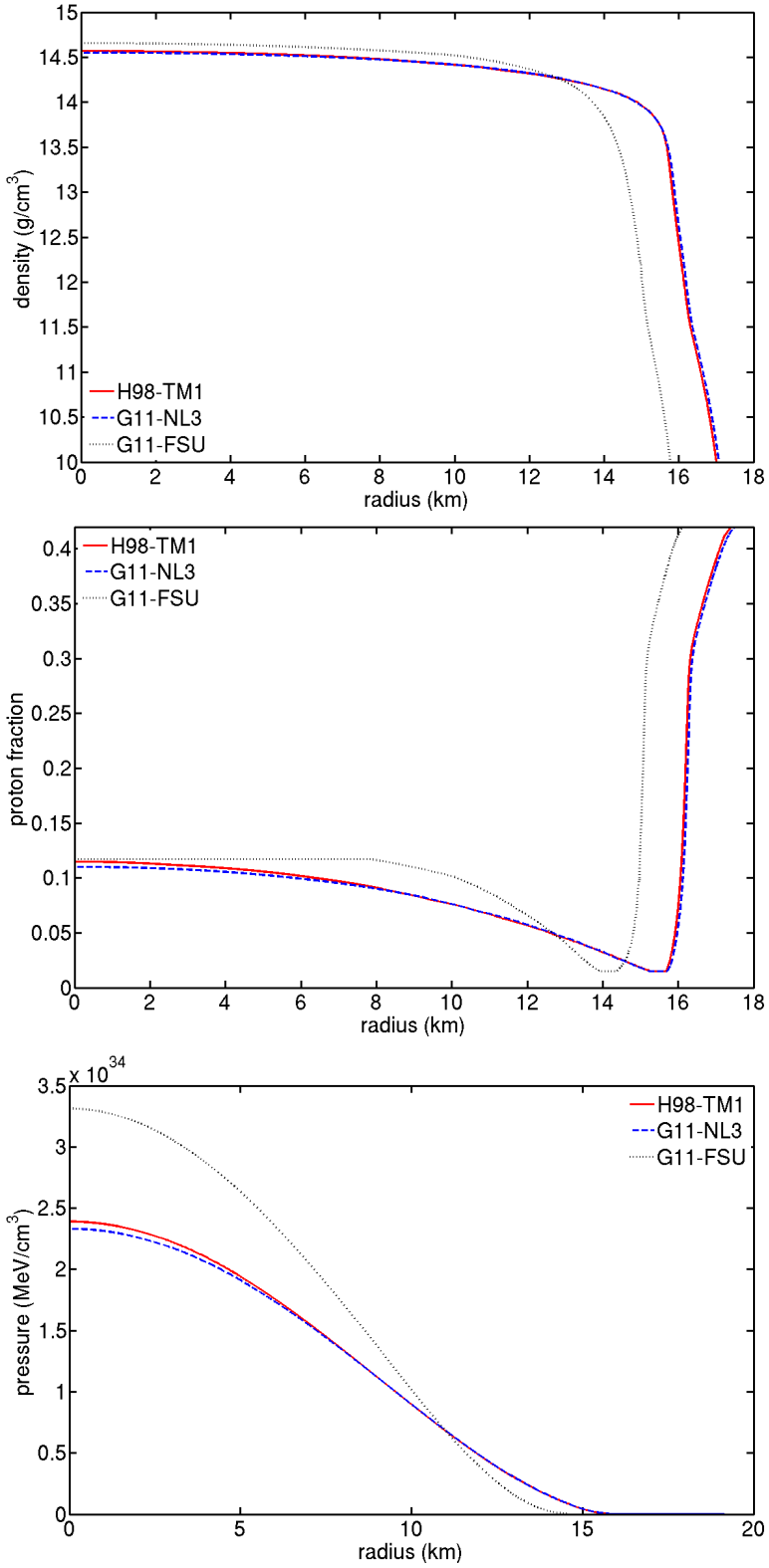


Figure 4.3: Radial profiles for the log density (top), the proton fraction (middle) and the pressure (bottom), for the neutron stars constructed for each EOS.

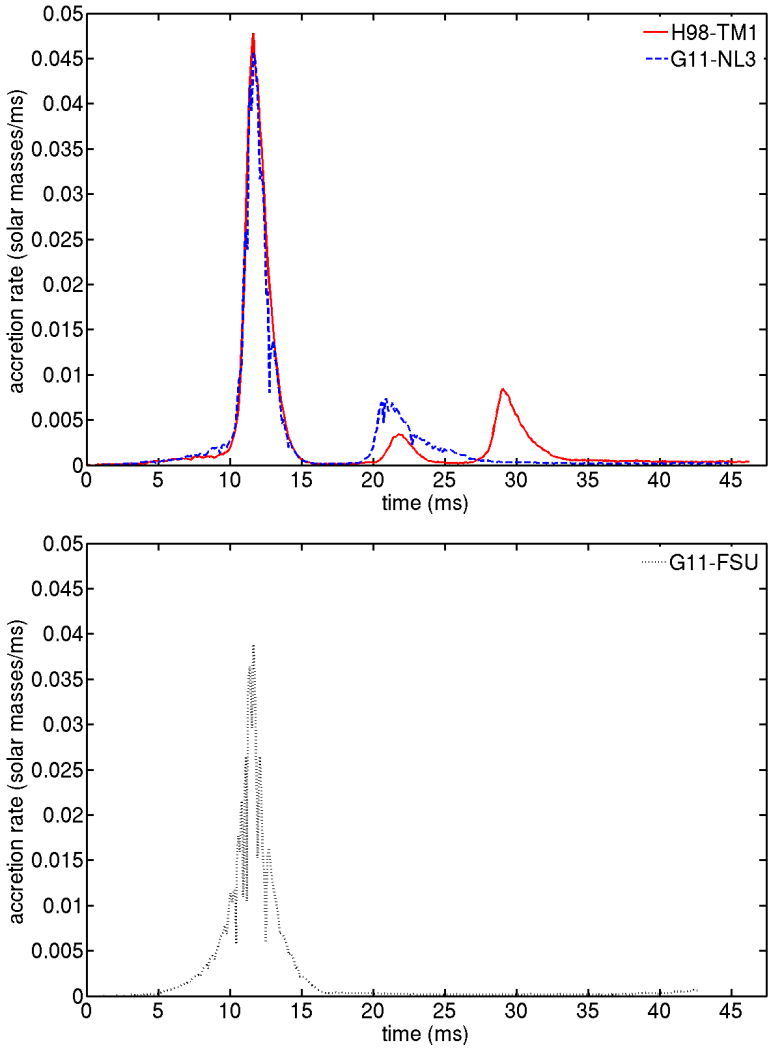


Figure 4.4: Accretion rate as a function of time, for H98-TM1 and G11-NL3 (top) and G11-FSU (bottom), for the $q = 3$ cases.

exactly to the Roche limit of the system, using the equation for a liquid satellite:

$$d = 2.44 R_{\rm NS} \left(\frac{M_{\rm BH}}{M_{\rm NS}} \right)^{\frac{1}{3}}. \tag{4.10}$$

If we use the estimate $R_{\rm NS} = 17$ km, then $d \sim 60$ km. Using eq. (15) of Artemova *et al.* (1996), we note that $R_{\rm ISCO} \sim 30$ km for a 4.8 $M_\odot$ BH (before accretion).

For both the H98-TM1 and G11-NL3 NSs, the first episode lasts around 5 ms, during which $\sim 80\%$ of their mass is lost (of which about 75% is accreted). The NS remnant is then flung out on a large elliptical orbit – an effect described as 'orbital widening' in Ruffert and Janka (2010) – with major axis greater than 140 km. The H98-TM1 is slung the furthest, and survives for two further mass transfer episodes, while the G11-NL3 NS survives for only one. We see from Figs. 4.6 and 4.7 that these episodes correspond exactly with the points in the

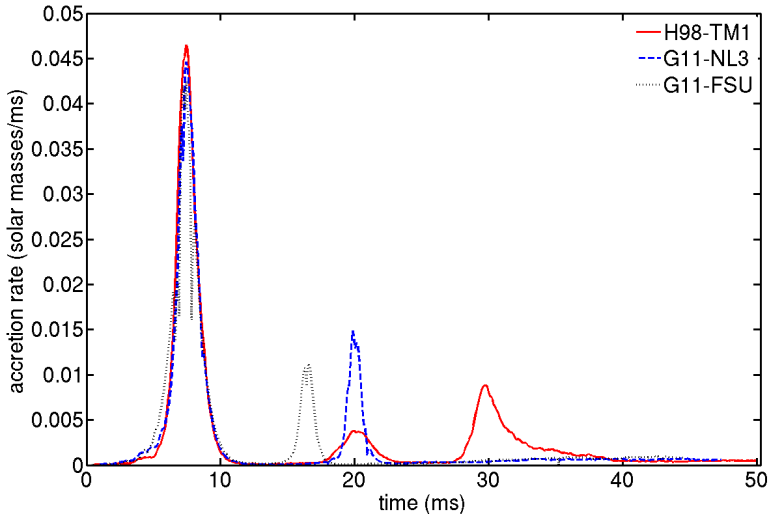


Figure 4.5: Accretion rate as a function of time, for the $q = 5$ cases.

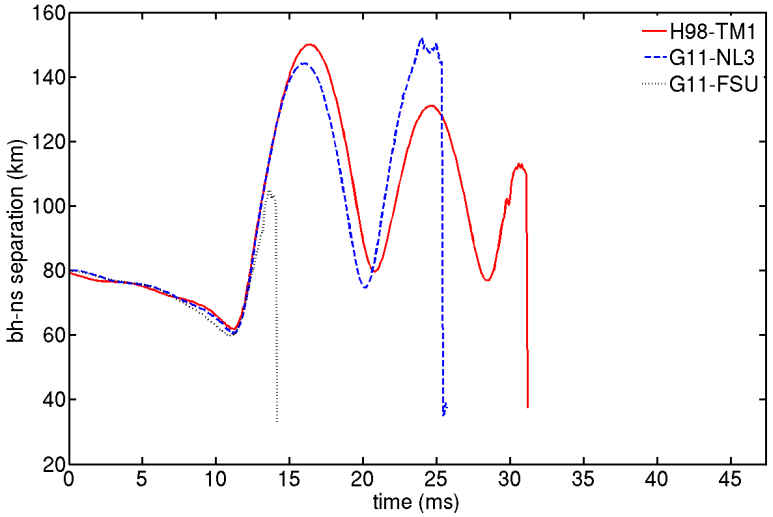


Figure 4.6: BH-NS separation as a function of time, for the $q = 3$ cases.

elliptical orbit when the BH-NS separation is at a (local) minimum.

In contrast to this, the G11-FSU NS loses almost all of its mass in a single episode lasting around 10 ms (of which about 75% is accreted). The mass transfer is also much less stable (as expected for a softer EOS), as shown by the oscillations in Fig. 4.4.

$q = 5$: these results are qualitatively similar to the $q = 3$ case. The first mass transfer episode now occurs at a distance of ~ 70 km. Again, this corresponds exactly to the Roche limit of the system, $d \sim 70$ km. For these cases, $R_{\rm ISCO} \sim 50$ km for a 8 $M_\odot$ BH.

At the first episode, the H98-TM1 and G11-NL3 lose around 60% of their mass, while the G11-FSU NS loses over 70%. In all cases, around 80% of this lost mass is accreted. One key difference between these results and the $q = 3$ case is that *all* the stars now undergo orbital

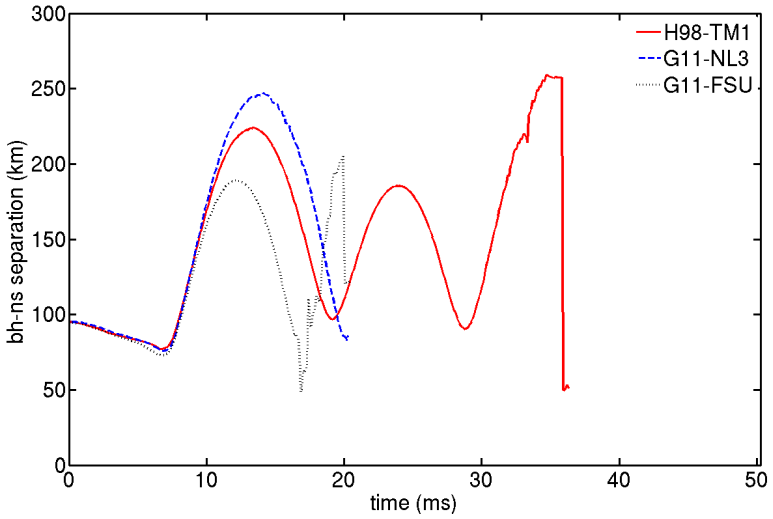


Figure 4.7: BH-NS separation as a function of time, for the $q = 5$ cases.

widening after the first mass transfer episode. Additionally, it is now the G11-NL3 NS which is flung the furthest, to a separation of nearly 250 km. The two G11 NSs survive for one further mass transfer event, while the H98-TM1 NS survives two.

Discussion

It has been suggested that the magnitude of the orbital widening effect depends on the accuracy with which the simulation code can model relativistic effects (e.g. Ruffert and Janka, 2010; Duez *et al.*, 2010). In purely Newtonian simulations, the NS can survive the initial mass transfer and form a self-bound 'mini'-NS, which is flung on a large, elliptical orbit. The orbital separation then decreases again – due to both the orbital shape and the angular momentum loss through gravitational wave emission – potentially resulting in multiple cycles of mass transfer and orbit widening (e.g. Janka *et al.*, 1999; Rosswog *et al.*, 2004).

However, this behaviour has not been observed in relativistic simulations, where the NS tends to be disrupted over a single mass transfer event (e.g. Duez *et al.*, 2010). Miller (2005) posited that this was due to the existence of the innermost stable circular orbit (ISCO), a relativistic effect with no Newtonian analog: the NS would more likely be absorbed through the ISCO than be tidally disrupted. This has also been found by previous pseudo-Newtonian studies, where the ISCO is approximated using a GR-mimicking potential (e.g. Rosswog, 2005; Ruffert and Janka, 2010). They found some orbital widening after the initial mass transfer, but a mini-NS does not form and there is generally no episodic mass transfer.

Despite the use of a GR-mimicking potential in this work (described in Section 3.3.1), it is clear from Figs. 4.4 to 4.7 that most of our results display the orbital widening and episodic mass transfer associated with purely Newtonian simulations. This can also be seen in the density slices given in Figs. 4.8 to 4.11. There are two possible explanations for these results.

The first explanation is that the NS is not instantly absorbed: the Roche limit of the system

– and hence the separation at which the NS is first tidally disrupted – is 20-30 km greater than $R_{\rm ISCO}$ in all cases. The inverse of this reasoning could also explain why orbital widening and episodic mass transfer was not observed in previous studies. For example, the pseudo-Newtonian study of Rosswog (2005) investigates H98-TM1, but their initial conditions involve significantly higher mass ratios (note that they define mass ratio as $M_{\rm NS}/M_{\rm BH}$, the reciprocal of our definition). This increases $R_{\rm ISCO}$, increasing the likelihood that the NS will be absorbed. Additionally, they do not take into account BH spin, which has a similar effect: without spin, $R_{\rm ISCO}$ would be 42 km and 70 km for our $q = 3$ and $q = 5$ cases respectively, and hence the NS would have crossed the ISCO before tidal disruption. The relativistic results of Duez *et al.* (2010) also investigate H98-TM1, with a mass ratio of $q = 3$ and a BH spin of $s = 0.5$ (i.e. the same conditions as our H98-TM1-3 case), but still do not find this orbital widening. However, we note that it is difficult to compare these results directly, because of the complications of defining masses and distances in GR physics (as opposed to Newtonian physics, where they are uniquely defined). Consequently, they find that the first mass transfer happens at ~ 40 km (rather than our ~ 60 km), well within the reach of the ISCO.

The second explanation is this behaviour could have some dependence on the stiffness of the EOS. We see evidence of this in our results: the soft G11-FSU NS undergoes the smallest amount of orbital widening for both mass ratios. This EOS dependence has been observed previously in purely Newtonian results, where NSs with stiffer EOSs are more likely to survive longer (Rosswog *et al.*, 2004). This could also explain why orbital widening was not generally observed by Ruffert and Janka (2010), as they use a $\Gamma = 2$ polytropic EOS, which is comparatively soft. We note that the soft G11-FSU-3 result is most similar to their results. Additionally, Deaton *et al.* (2013) investigate a mass ratio $q = 4$ and a high BH spin of $q = 0.9$. In Newtonian/pseudo-Newtonian results, these conditions would be more conducive to orbital widening. Again, they do not observe this, but this could be explained by their use of the softer Lattimer-Swesty EOS. Hence, a more thorough exploration of the parameter space – with both pseudo-Newtonian and relativistic codes – is required to narrow down the effect of the EOS on the orbital widening/episodic mass transfer.

4.4.3 Final State: the Disk, the BH and Ejecta

The properties of the system 42 ms after the first major mass transfer event are given in Table 4.4. At this point, accretion has slowed down and the disk is in a quasi-equilibrium state in all cases. $M_{\rm avail}$ is the amount of matter surrounding the BH that has not been absorbed or ejected, and hence available to form a disk. This is important in the context of GRBs, as the disk properties determine how much energy is available to power a relativistic jet via neutrino processes. However, note that $M_{\rm avail}$ is found by summing the gas mass over the entire domain, and is hence an upper bound on the disk mass. $M_{\rm unb}$ is the total unbound mass of the system, and hence an approximation for the amount of ejected mass. Mass is classed as 'unbound' if it has sufficient energy to overcome the gravitational potential energy at its location. Finally, Table 4.4 also lists the maximum density and temperature in the disk, the density-weighted average temperature, and the BH mass.

Model	$M_{\rm avail}$ $M_\odot$	$M_{\rm avail}$ % of $M_{\rm NS}$	$\rho_{\rm max}$ 10^{14} g cm^{-3}	$k_B T_{\rm max}$ MeV	$\langle k_B T \rangle$ MeV	$M_{\rm BH}$ $M_\odot$	$M_{\rm unb}$ $M_\odot$
H98-TM1-3	0.48	30%	12.0	9.9	0.25	6.06	0.01
G11-NL3-3	0.40	25%	11.7	10.2	0.19	6.06	$< 10^{-2}$
G11-FSU-3	0.40	25%	11.7	12.2	0.24	6.03	$< 10^{-2}$
H98-TM1-5	0.47	31%	11.6	10.1	0.078	9.26	0.03
G11-NL3-5	0.50	31%	11.9	10.4	0.37	9.16	$< 10^{-2}$
G11-FSU-5	0.42	26%	11.8	11.4	0.35	9.22	$< 10^{-2}$

Table 4.4: Properties of the system 42 ms after the first major mass transfer event, when the disk is in a quasi-equilibrium state. $M_{\rm avail}$ is the amount of mass available to form a disk (i.e. NS matter that has not been accreted or ejected). $\rho_{\rm max}$ and $k_B T_{\rm max}$ are the maximum density and temperature of the disk, and $\langle k_B T \rangle$ is the density-weighted average temperature. Finally, we include the mass of the black hole, $M_{\rm BH}$, and the unbound mass (an approximation of the ejected mass), $M_{\rm unb}$.

Figs. 4.8 to 4.11 show the density distributions in the orbital plane. Figs. 4.8 to 4.10 show the evolution of the system, for all three EOSs, for the $q = 3$ cases. Fig. 4.11 shows only the quasi-equilibrium disk at late times, for all three EOSs, for the $q = 5$ cases. To aid comparisons, the colour bar has been kept fixed for all density plots.

Results

In all cases, the NS spreads out into a large tidal tail, which reaches over 500 km in radius. Eventually, this tail falls back towards the BH, wrapping around it and colliding with itself, leading to the formation of an accretion disk. This process happens slowest for the H98-TM1 cases and fastest for the G11-FSU cases. This can be seen in Figs. 4.6 and 4.7, as the time taken for accretion disk formation is related to how long the NS takes to become completely tidally disrupted (and hence when these graphs are cut-off). It can also be seen in the central panels of Figs. 4.8 to 4.10 (depicting roughly equal times of $\sim$ 26-28 ms after the first major mass transfer event): the core of the H98-TM1 NS is still visible, while the G11-FSU NS has already completely shredded and formed an accretion disk. From Table 4.4, we see that the amount of matter surrounding the BH that is available to form this disk does not depend strongly on the EOS, with all cases leaving $\sim$ 0.4 - 0.5 $M_\odot$.

However, the *properties* of this disk are more dependent on the EOS. In the $q = 3$ cases, the H98-TM1 NS forms a smaller, higher-density disk than both of the G11 NSs (as shown in the lower panels of Figs. 4.8 to 4.10). Although the density-weighted temperatures are similar, $\sim 0.2 - 0.25$ MeV, the G11-FSU disk has the highest maximum temperature. In the $q = 5$ cases, the H98-TM1 NS still forms the smallest disk, but the two G11 disks have higher-density central regions (as shown in Fig. 4.11). Additionally, the density-weighted average temperature of the H98-TM1 disk is significantly lower than for the two G11 disks. However, the G11-FSU disk still produces the highest maximum temperature. We find that all the disks are also fairly thick, with $H_{\rm disk}/R_{\rm disk} \sim$ 0.2 - 0.4. Finally, we note that the unbound mass has

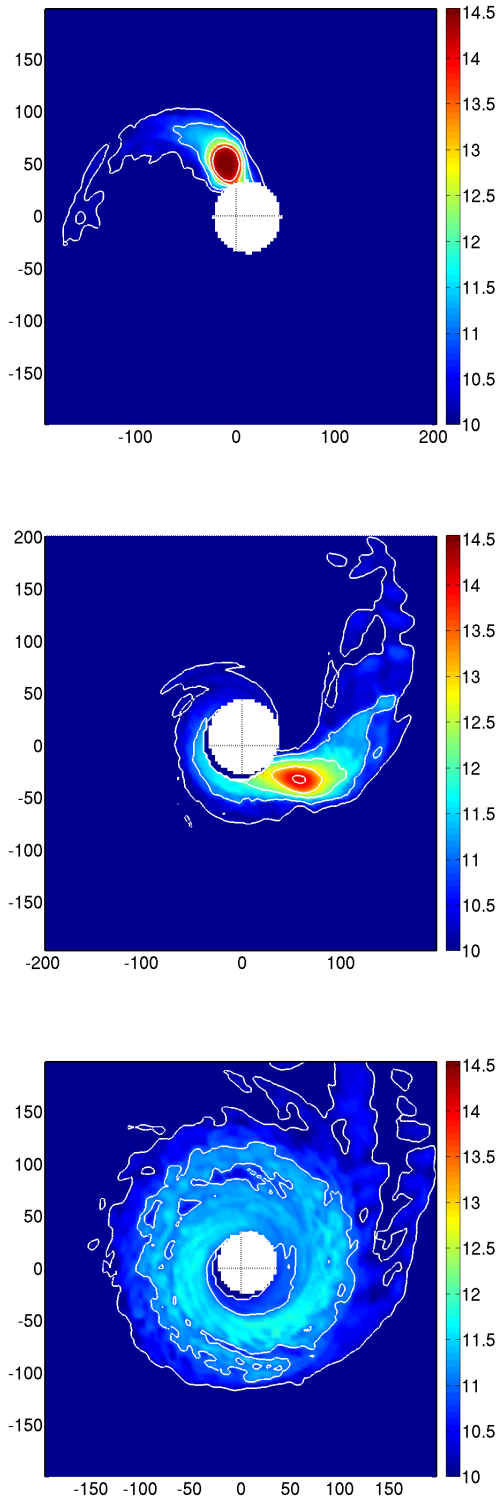


Figure 4.8: The mass density distributions in the orbital plane for H98-TM1-3, at $t = 8.33$ ms (top), $t = 28.0$ ms (middle) and $t = 37.6$ ms (bottom). The white lines are logarithmically spaced contours from 10^{10} - 10^{15} g cm^{-3}. The white circle at the centre represents the BH.

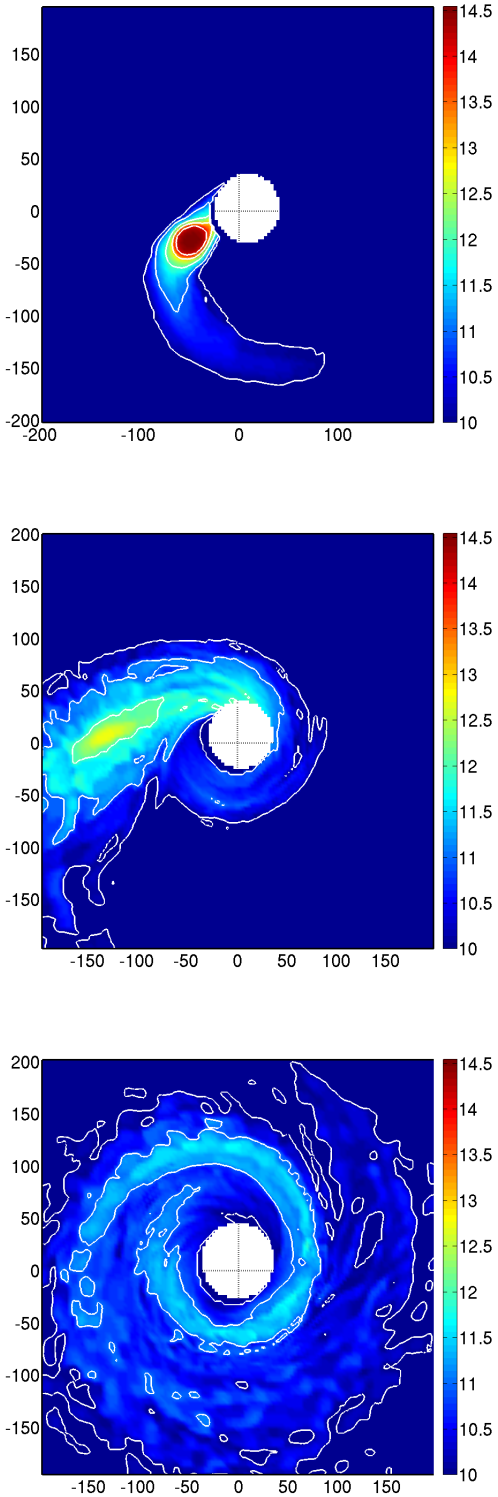


Figure 4.9: The mass density distributions in the orbital plane for G11-NL3-3, at $t = 8.53$ ms (top), $t = 26.3$ ms (middle) and $t = 35.2$ ms (bottom). The white lines are logarithmically spaced contours from 10^{10} - 10^{15} g cm^{-3}. The white circle at the centre represents the BH.

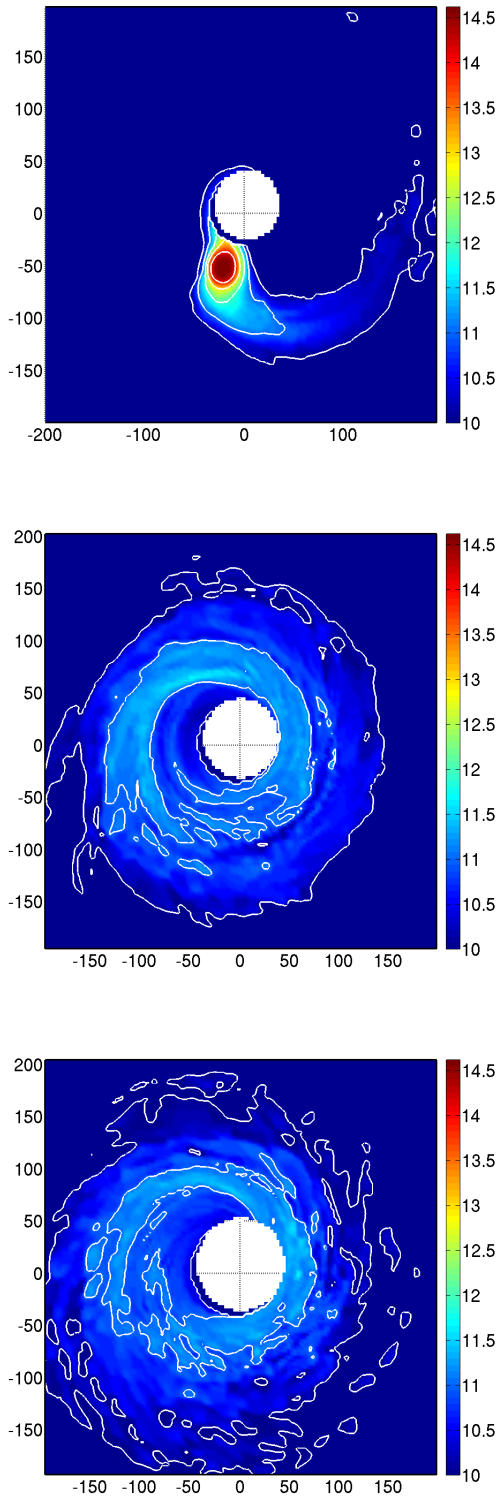


Figure 4.10: The mass density distributions in the orbital plane for G11-FSU-3, at $t = 8.86$ ms (top), $t = 28.6$ ms (middle) and $t = 38.2$ ms (bottom). The white lines are logarithmically spaced contours from 10^{10} - 10^{15} g cm^{-3}. The white circle at the centre represents the BH.

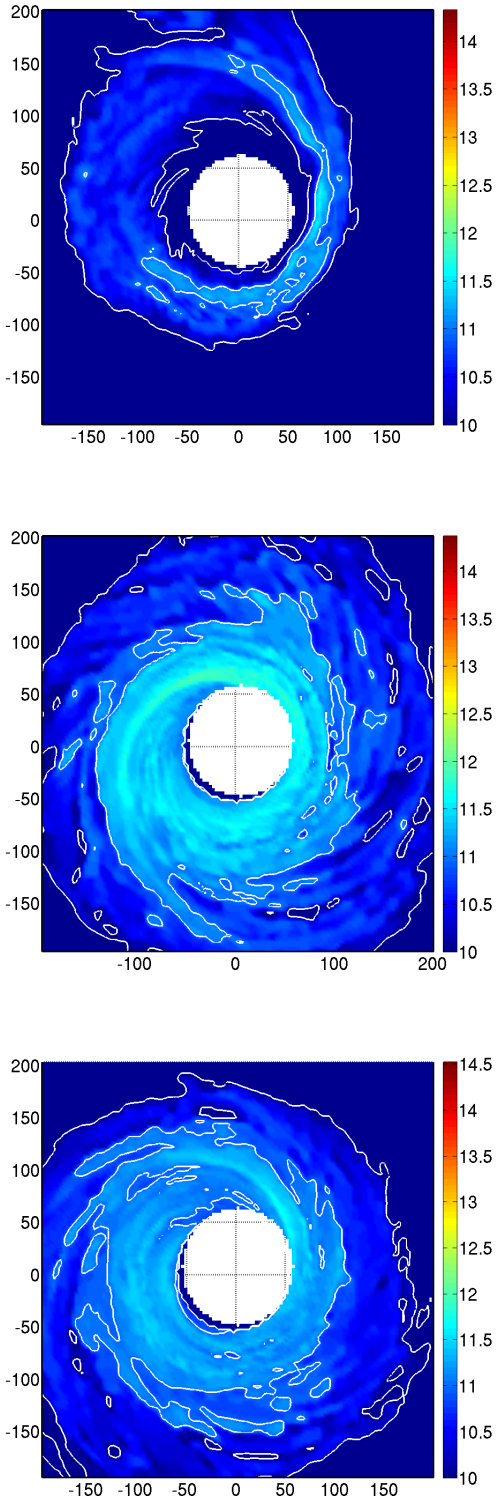


Figure 4.11: Late-time mass density distributions in the orbital plane, after the NS has been completely tidally shredded, for the $q = 5$ cases. H98-TM1-5 is shown at $t = 41.1$ ms (top), G11-NL3-5 at $t = 38.7$ ms (middle) and G11-FSU-5 at $t = 36.5$ ms (bottom). The white lines are logarithmically spaced contours from 10^{10} - 10^{15} g cm^{-3}. The white circle at the centre represents the BH.

Model	$L_{\nu,\max}$ erg s^{-1}	$\dot{M}_{\max}$ $M_\odot$ ms^{-1}	ε
H98-TM1-3	8.5 $\times 10^{53}$	0.046	0.01
G11-NL3-3	2.0 $\times 10^{54}$	0.043	0.02
G11-FSU-3	7.4 $\times 10^{54}$	0.037	0.11
H98-TM1-5	1.3 $\times 10^{54}$	0.047	0.01
G11-NL3-5	2.8 $\times 10^{54}$	0.042	0.03
G11-FSU-5	1.7 $\times 10^{55}$	0.045	0.21

Table 4.5: Properties of the neutrino emission. $L_{\nu,\max}$ is the peak neutrino luminosity (for all species of neutrino), $\dot{M}_{\max}$ is the maximum accretion rate and $\varepsilon = L_{\nu,\max}/(\dot{M}_{\max}c^2)$ is the accretion efficiency.

some EOS dependence, because only H98-TM1 produces a significant amount.

Discussion

For polytropic EOSs, Duez *et al.* (2010) find that a softer EOS – and hence higher NS compactness – results in a smaller tidal tail, faster fall-back towards the BH and a lighter disk. This agrees with our results for the softest EOS, G11-FSU. However, they also found that H98-TM1 resulted in the most rapid fall-back, faster than even their softest polytrope. It is is plausible that the inclusion of realistic microphysics systematically speeds up the process over polytropic EOSs; further relativistic studies with the G11-NL3 and G11-FSU EOSs (or other realistic EOSs) are required to confirm whether this is the case.

Previous studies have found a wide range of values for the amount of mass left around the BH, ranging from $\sim 10^{-4} M_\odot$ – likely below the threshold for GRB production – to $\sim 0.5 M_\odot$. Our estimates for this value are the same order of magnitude as previous relativistic studies using similar parameters. Duez *et al.* (2010) – which uses H98-TM1 but with no neutrino cooling – found 0.1 $M_\odot$ for $q = 3$ and $s = 0.5$. Deaton *et al.* (2013) – which uses the Lattimer-Swesty EOS with neutrino cooling – found 0.3 $M_\odot$, for $q = 4$ and $s = 0.9$.

We note that our results are higher than these values, but this can be explained by a number of factors. The use of a stiffer EOS increases the disk mass, as described above. H98-TM1 and G11-NL3 are both stiffer than the Lattimer-Swesty EOS; G11-FSU is softer at high densities but stiffer at lower densities. (See fig. 11 of Shen *et al.* (2011a) for a comparison of all four.) We also include neutrino effects; Deaton *et al.* (2013) ran simulations both with and without neutrino cooling, and found that the latter resulted in a 40% lower (baryonic) disk mass. Finally, as documented in Ruffert and Janka (2010), pseudo-Newtonian simulations tend to result in systematically higher disk masses than general relativistic results.

4.4.4 Neutrino Emission

We now consider the neutrino emission, which is powered by the thermal energy of the disk. This is important in the context of GRBs, because neutrino-antineutrino annihilation and

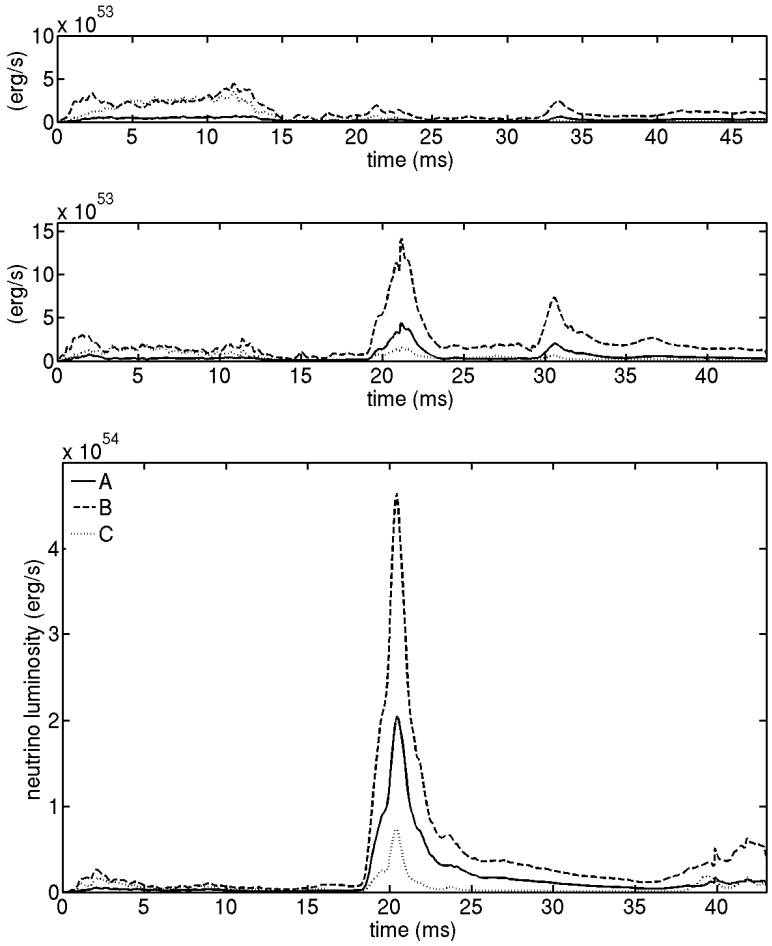


Figure 4.12: Neutrino luminosity as a function of time, for the $q = 3$ case, for H98-TM1 (top), G11-NL3 (middle) and G11-FSU (bottom). The solid line 'A' is for electron neutrinos, the dashed line 'B' is for electron anti-neutrinos, and the dotted line 'C' is for all heavier neutrinos.

subsequent pair production can potentially power a relativistic jet.

The neutrino luminosity for ν_e, $\bar{\nu}_e$ and ν_x (all heavier neutrinos) are given in Figs. 4.12 ($q = 3$) and 4.13 ($q = 5$). Note that these graphs have been sized so that the scale of the y-axis remains the same. The peak neutrino luminosity (for all species) is given in Table 4.5. We also include the peak accretion rate, $\dot{M}_{\rm max}$, and use this to estimate the accretion efficiency, $\varepsilon = L_{\nu,\rm max}/(\dot{M}_{\rm max}c^2)$. The cumulative neutrino energy is given in Figs. 4.14 ($q = 3$) and 4.15 ($q = 5$). Recall that all neutrino quantities were determined using the neutrino leakage scheme described in Ruffert *et al.* (1996) (and summarised in Section 3.2.1).

Results

In all cases, there is a small amount of neutrino emission for the first $\sim$ 5-10 ms of the simulation. However, this is a result of unphysical shock heating, due to the neutron star colliding with the ambient background matter ($\sim 10^8$ g cm^{-3}) required for the code to run.

G11-NL3: for both mass ratios, there is a neutrino pulse $\sim$ 5 ms after the first mass

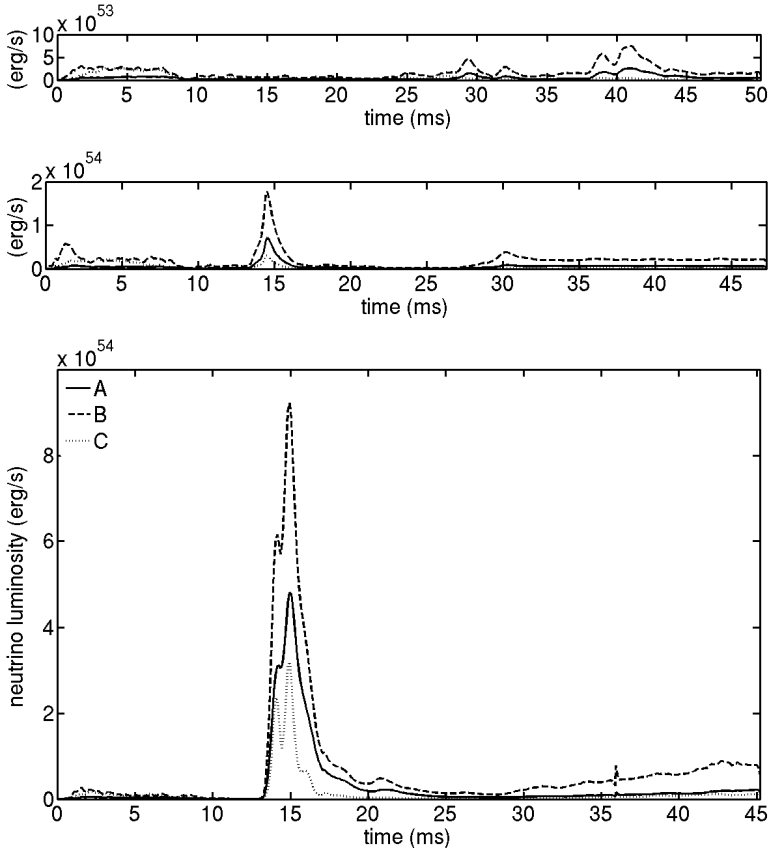


Figure 4.13: Neutrino luminosity as a function of time, for the $q = 5$ case, for H98-TM1 (top), G11-NL3 (middle) and G11-FSU (bottom). The solid line 'A' is for electron neutrinos, the dashed line 'B' is for electron anti-neutrinos, and the dotted line 'C' is for all heavier neutrinos.

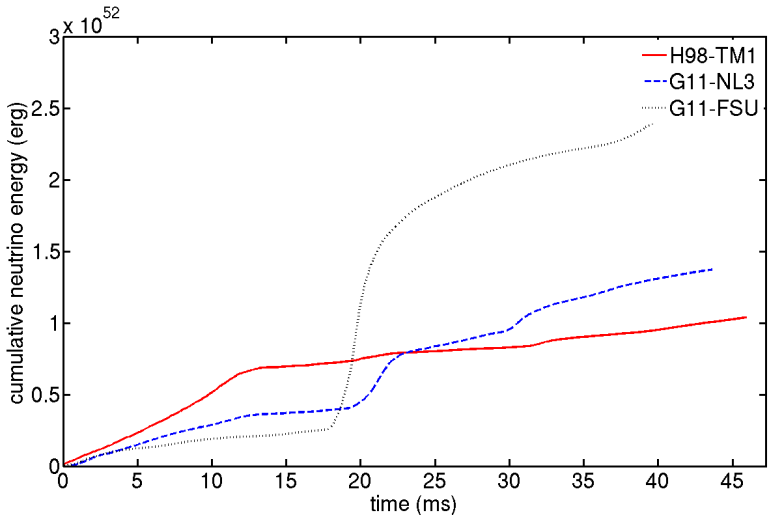


Figure 4.14: Cumulative ν energy as a function of time, for the $q = 3$ case.

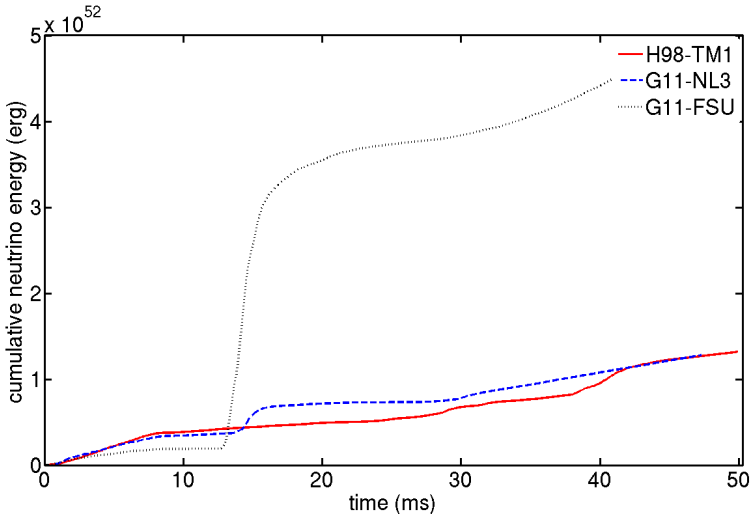


Figure 4.15: Cumulative ν energy as a function of time, for the $q = 5$ case.

transfer episode, lasting around 3-5 ms. These have peak luminosity of $\sim 10^{54}$ erg s^{-1}. This is followed by a lower luminosity pulse ~ 10 ms after the second mass transfer episode. At late times, the neutrino luminosity is steady at $\sim 10^{53}$ erg s^{-1}.

G11-FSU: again, for both mass ratios, there is a neutrino pulse ~ 5 ms after the first mass transfer episode. These are around five times more luminous than for G11-NL3, even reaching $\sim 10^{55}$ erg s^{-1} in the $q = 5$ case. There is no secondary mass transfer episode in the $q = 3$ case, and hence no secondary neutrino pulse. In the $q = 5$ case, the second mass transfer episode happens relatively quickly – within 10 ms of the first – which appears to result in a 'double peak'. This is most clearly visible in the ν_x-emission. At late times, the neutrino luminosity is $\sim 10^{53}$ erg s^{-1}, but is increasing steadily.

H98-TM1: for the $q = 3$ case, there are two low-luminosity pulses corresponding to the first two mass transfer episodes. However, these are less luminous than the emission due to the unphysical heating described above. For the $q = 5$ case, there is no clear pulse visible at early times, but there are a series of mini-pulses at late times. Towards the end of the simulation time, the neutrino luminosity appears to be steady at $\sim 10^{53}$ erg s^{-1}.

The consequences of these results can be seen in Figs. 4.14 and 4.15: there is a sharp increase in the cumulative neutrino energy released by G11-FSU during the first neutrino pulse, after which the energy increases at a similar rate for all three EOSs.

Discussion

The neutrino luminosities obtained are higher than previous pseudo-Newtonian studies: Rosswog (2005) find a best-case luminosity of $\sim 10^{51}$ erg s^{-1}, and Setiawan *et al.* (2006) find $\sim 10^{53}$ erg s^{-1}. However, our results are comparable with the relativistic study of Deaton *et al.* (2013), which finds an initial luminosity of $\sim 10^{54}$ erg s^{-1}. These high luminosities are likely due to the generally larger disk masses (recall that Deaton *et al.* (2013) find a disk mass of 0.3 $M_\odot$). This is clear from Table 4.4, as the efficiencies ($\sim$ 1 -10%) are comparable

with what has been obtained previously: a few percent by Setiawan *et al.* (2006) and 0.1% by Deaton *et al.* (2013). We also note that we approximate the viscosity using the method of Setiawan *et al.* (2006), which was found to increase the neutrino luminosity by 30-40%.

It is clear from these results that the neutrino luminosity depends strongly on the EOS: G11-FSU resulted in the most luminous pulses despite having the smallest disk mass for both mass ratios. G11-NL3 resulted in more luminous pulses than H98-TM1, but the cumulative neutrino energy is comparable for the simulated time, especially for the $q = 5$ case. Recall from Section 4.1.3 that, unlike H98-TM1, both the G11 EOSs use a.) exact solutions to the RMF equations and b.) thousands of species of heavy nuclei to model the low-density regime. The main consequence of these differences appears to be the production of sharp, luminous neutrino pulses, which do not appear as clearly in the H98-TM1 results.

Finally, we note that all cases sustain a luminosity of at least $\sim 10^{53}$ erg s^{-1} for the simulated time, which is compatible with the required luminosity for GRB production. However, the sharp pulses produced by the G11 EOSs are a better fit for the burst-like quality of SGRBs.

4.5 Conclusions

In this chapter, we presented the first BH-NS merger simulations to use the realistic EOSs of G. Shen *et al.* (2011b, G11-NL3) and G. Shen *et al.* (2011a, G11-FSU). For reference, we compared these with simulations that used the EOS of H. Shen *et al.* (1998a b, H98-TM1). We found that the outcome of the merger is strongly influenced by the EOS. Specifically:

- The number of mass transfer episodes depend on the EOS: the H98-TM1 NS consistently survived the longest, despite not being the stiffest EOS.
- The resultant disk mass does not depend strongly on the EOS, with all runs producing $\sim 0.4 - 0.5\ M_{\odot}$. However, the disk properties do: H98-TM1 produces the most compact disk, while G11-FSU results in the most extreme temperatures.
- Only H98-TM1 produces non-negligible ejecta.
- The G11 EOSs produce sharp neutrino pulses, with G11-FSU reaching a peak luminosity of $\sim 10^{55}$ erg s^{-1} in the $q = 5$ case. However, all cases sustain a luminosity of at least $\sim 10^{53}$ erg s^{-1}, which is sufficient for GRB production.

We also observe 'orbital widening' of the NS, which was previously thought to be suppresed by the use of GR-mimicking potentials. We posit that this effect is dependent on the EOS, as well as other parameters such as the black hole mass and spin. Further studies – with both pseudo-Newtonian and relativistic codes – are required to narrow down its cause.

Chapter 5
Modelling Magnetic Fields (1D)

We now turn to the second main project of this thesis: upgrading our simulation code, Charybdis, to model magnetic field effects.

The first step is to modify the reconstruct and solve stages of the RSA stategy (described in Chapter 2) to handle the equations of magnetohydrodynamics. In this chapter, we implement four 1D MHD reconstruction algorithms in a test code, Magneto, and present results from several 1D tests to compare them. We also implement the MHD Riemann solver of Miyoshi and Kusano (2005).

5.1 Introduction

5.1.1 The MHD Equations

To include magnetic field effects in Charybdis, we must first upgrade the hydrodynamics engine to solve the equations of *magneto*hydrodynamics (MHD). These combine the Euler equations with Maxwell's equations of electromagnetics, in order to determine the dynamics of a compressible and conducting fluid under the influence of magnetic fields.

This upgrade involves two main tasks, as pointed out by e.g. Gardiner and Stone (2005). The first task is that we must extend the reconstruction and Riemann solver algorithms to include the three magnetic field components: B_x, B_y and B_z. Helpfully, the ideal 1D MHD equations can be written in the same conservative, differential form as the 1D Euler equations:

$$\frac{\partial}{\partial t}\begin{bmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ \mathcal{E} \\ B_x \\ B_y \\ B_z \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} \rho v_x \\ \rho v_x^2 + p_T - B_x^2 \\ \rho v_x v_y - B_x B_y \\ \rho v_x v_z - B_x B_z \\ v_x(\mathcal{E} + p_T) - B_x(\mathbf{v}\cdot\mathbf{B}) \\ 0 \\ -E_z = v_x B_y - v_y B_x \\ E_y = v_x B_z - v_z B_x \end{bmatrix} = 0, \tag{5.1}$$

where we have followed the component form of e.g. Stone *et al.* (2008). Further notes:

- p_T is the *total* pressure, such that $p_T = p_G + \frac{1}{2}|\mathbf{B}|^2$. p_G is the usual gas (or thermal) pressure, previously denoted as just p, while $\frac{1}{2}|\mathbf{B}|^2$ is the magnetic pressure.
- $\mathbf{E} = (E_x, E_y, E_z)$ is the electric field, and is given by Ohm's law: $\mathbf{E} + \mathbf{v} \times \mathbf{B} = 0$.
- We use SI units for all the magnetic field components, as opposed to CGS units, which would introduce a factor of $1/\sqrt{4/\pi}$ (see e.g. eq. (4) of Balsara and Spicer, 1999).
- Finally, we assume an ideal EOS for the tests in this chapter: the gas pressure is given by $p_G = (\gamma - 1)\ (\mathcal{E} - \frac{1}{2}\rho|\mathbf{v}|^2 - \frac{1}{2}|\mathbf{B}|^2)$, where γ is some constant adiabatic index.

The second task is that we must deal with the divergence-free condition, $\nabla \cdot \mathbf{B} = 0$. This is one of Maxwell's equations, and so must be maintained at all times. This is simple in one dimension: it is maintained automatically if the parallel component of the magnetic field (B_x) is kept uniform in space and constant in time. Uniformity can be set with the initial conditions, while constancy is guaranteed by the MHD equations: the flux corresponding to B_x is zero, and hence $\partial B_x/\partial t = 0$. This condition is considerably less simple to maintain in more than one dimension, but we will withhold this discussion until Chapter 6.

5.1.2 Miscellany

Conserved, Primitive and Characteristic Variables

Before we continue, we will need to distinguish between three types of variables. *Conserved* variables are denoted by the vector $\mathbf{u}$, and represent the quantities that are conserved by the system of conservation equations. For MHD:

$$\mathbf{u} = [\rho, \rho v_x, \rho v_y, \rho v_z, \mathcal{E}, B_y, B_z], \tag{5.2}$$

where we have dropped the B_x component, because this is not evolved in a 1D system. However, many algorithms work in the *primitive* variables, denoted by the vector $\mathbf{v}$:

$$\mathbf{v} = [\rho, v_x, v_y, v_z, p_G, B_y, B_z]. \tag{5.3}$$

We use the same symbol for the primitive variables and the velocity, but it should be clear which we are referring to from the context. Finally, we also work with the *characteristic* variables, denoted by the vector $\mathbf{w}$. Recall from Section 2.4.1 that:

$$\mathbf{w} = \mathbf{L}\mathbf{v}, \tag{5.4}$$

where $\mathbf{L}$ is the matrix of left eigenvectors of the flux Jacobian matrix $\mathbf{A}$. Note that we now define this matrix as $\mathbf{A} = \partial\mathbf{f}/\partial\mathbf{v}$, as opposed to the definition $\mathbf{A} = \partial\mathbf{f}/\partial\mathbf{u}$ used in Chapter 2.

For the algorithms and tests described in this chapter, we will start with the primitive variables $\mathbf{v}$. However, it is straightforward to convert between all three, assuming that we know the eigensystem of the matrix $\mathbf{A}$ (for $\mathbf{v} \leftrightarrow \mathbf{w}$) and the equation of state (for $\mathcal{E} \leftrightarrow p$) .

Test Code: Magneto

The algorithms described in Chapters 5 and 6 have all been implemented in a test code, MAGNETO. This code has been released freely under the GNU public licence, and – at time of writing – can be downloaded from `http://github.com/hsriskantha/magneto/`. Instructions on how to use it can be found in the `readmes` folder. It is designed to be clear and illustrative (rather than high performance), in order to aid understanding of the implemented algorithms, especially the new algorithm presented in the next chapter.

Whenever we need to refer to a subroutine or function in MAGNETO, we will do so using its unique five character keycode. The first three characters (letters) indicate the module name, while the last two characters (numbers) indicate the position within the code file. For example: `[FLU01]` is the first subroutine in the fluid module, `B-fluid-mod.f90`. A list of all subroutines and functions is given in `readmes/contents.txt`.

5.2 Upgrading the Reconstruct Stage

Recall that in our discretised 1D system, the density functions are stored as cell averages $\mathbf{U}_n^i$ (at time t_n). Using these cell averages, the aim of the reconstruct stage is to approximate the value of the original functions on either side of each cell interface, $(i*)$. These values, which we label $\mathbf{U}_{n*,L}^{i*}$ and $\mathbf{U}_{n*,R}^{i*}$, are then used as input for the Riemann solver.

In Section 2.4.5, we briefly introduced the Piecewise Parabolic Method of Colella and Woodward (1984), which uses a parabolic interpolation on the primitive variables. We could simply extend this approach – given by eq. (2.41) – to include interpolation of B_x, B_y and B_z. However, it has been suggested by e.g. Balsara (1998) that interpolation in the *characteristic* variables is better, because it provides stronger entropy enforcement than the interpolation on the primitive variables.[1] Of course, a disadvantage of this approach is that we need to determine the eigensystem for every cell, at each timestep. While this is straightforward to implement – the analytic eigensystem for ideal MHD is known, and given in Appendix A – it is computationally more expensive than algorithms which do not require this step.

To investigate this – and hence determine the best algorithm to use for subsequent investigations – we implement three algorithms that use such characteristic interpolation:

- a second-order reconstruction with steepening algorithm[2] by Balsara (1998) (which we refer to as **BAL**, and is implemented in `[REC02]`),
- a second-order piecewise linear reconstruction by Stone *et al.* (2008) (**PLM**, `[REC03]`),

[1] A brief explanation for this: the analytic solution of a system of hyperbolic equations has been shown to be total variation diminishing (TVD) (Balsara, 1998, and references therein). Balsara suggests that interpolation in the characteristic variables is a more natural numerical interpretation of this analytic result. Recall from Section 2.4.5 that this TVD property is desirable, as it reduces unphysical oscillations in our numerical results.

[2] Balsara (1998) also notes that characteristic interpolation produces a shallower representation of jumps in linearly degenerate fields – i.e. the Alfvén waves and the entropy wave – hence the need for a steepening algorithm.

- a third-order piecewise parabolic reconstruction by Stone *et al.* (2008) (**PPM**, `[REC04]`).

All three algorithms loosely follow the approach of Colella (1990), which we summarise below. We will compare these to the standard PPM algorithm, summarised in Section 2.4.5, which uses interpolation in the primitive variables. However, we implement the version developed by Colella and Sekora (2008) (CSK, `[REC05]`), which uses a different limiter to preserve accuracy at smooth extrema. Note that these approaches can be applied to *all* hyperbolic conservation laws, including pure hydrodynamics, but we focus on MHD.

5.2.1 The Theory

We start by assuming that the value of the original function is equal to the cell average at the cell centre. We can then interpolate to the cell interfaces, using spatial and temporal derivatives. For example, for the left state (using primitive variables):

$$\mathbf{V}^{i*}_{n*,L} = \mathbf{V}^{i}_{n} + \frac{\Delta x}{2}\frac{\partial \mathbf{v}}{\partial x} + \frac{\Delta t}{2}\frac{\partial \mathbf{v}}{\partial t} \tag{5.5}$$

$$= \mathbf{V}^{i}_{n} + \frac{\Delta x}{2}\frac{\partial \mathbf{v}}{\partial x} - \frac{\Delta t}{2}\frac{\partial \mathbf{f}}{\partial x} \tag{5.6}$$

$$= \mathbf{V}^{i}_{n} + \left(\frac{\Delta x}{2} - \frac{\Delta t \mathbf{A}}{2}\right)\frac{\partial \mathbf{v}}{\partial x}, \tag{5.7}$$

where we have used the conservation law $\partial \mathbf{v}/\partial t = -\partial \mathbf{f}/\partial x$ in eq. (5.6), and the flux Jacobian matrix $\mathbf{A} = \partial \mathbf{f}/\partial \mathbf{v}$ in eq. (5.7). Note that we use a capital $\mathbf{V}$ for discretised values. Similarly, we construct the right state from the other side:

$$\mathbf{V}^{i*}_{n*,R} = \mathbf{V}^{i+1}_{n} - \left(\frac{\Delta x}{2} - \frac{\Delta t \mathbf{A}}{2}\right)\frac{\partial \mathbf{v}}{\partial x}. \tag{5.8}$$

For linear systems, the spatial derivative $\partial \mathbf{v}/\partial x$ can simply be approximated by the finite difference $\Delta \mathbf{V}/\Delta x$. However, for strongly nonlinear systems – such as the MHD equations – Colella (1990) suggests two additional steps to make the algorithm more robust. The first step is to use intermediate 'arbitrary' *reference states* $\hat{\mathbf{V}}^{i*}_{L}$ and $\hat{\mathbf{V}}^{i*}_{R}$, which integrate the slope $\Delta \mathbf{V}$ over the domain of dependence of the cell interface $(i*)$. This domain is defined by the minimum and maximum eigenvalues (which represent the minimum and maximum wavespeeds). The second step is *characteristic tracing*, which corrects this integral by subtracting contributions from waves that do not reach the cell interface within time $\Delta t/2$. The derivation of these methods is outside the scope of this thesis, so we will skip straight to the algorithmic details.

5.2.2 The Algorithms

(a) Calculating the Eigensystem

We start by converting the cell-averaged conserved variables, $\mathbf{U}^{i}$, into the primitive variables, $\mathbf{V}^{i}$. Note that we have dropped the time index n for clarity. We then calculate the eigenvalues

λ_p (where $p \in \{1...7\}$), and the corresponding matrix of left and right eigenvectors, $\mathbf{L}(\mathbf{V}^i)$ and $\mathbf{R}(\mathbf{V}^i)$, using the analytic expressions given in Appendix A.

(b) Determining the Spatial Derivative

Next we approximate the slope $\Delta\mathbf{V}$, in order to determine the spatial derivative, $\Delta\mathbf{V}/\Delta x$. PLM and PPM first define the following differences:

$$\Delta\mathbf{V}^{L,i} = \mathbf{V}^i - \mathbf{V}^{i-1}, \tag{5.9}$$

$$\Delta\mathbf{V}^{R,i} = \mathbf{V}^{i+1} - \mathbf{V}^i, \tag{5.10}$$

$$\Delta\mathbf{V}^{C,i} = (\mathbf{V}^{i+1} - \mathbf{V}^{i-1})/2. \tag{5.11}$$

These differences are then projected onto the characteristic variables:

$$\Delta\mathbf{W}^{L,i} = \mathbf{L}(\mathbf{V}^i) \cdot \Delta\mathbf{V}^{L,i}, \tag{5.12}$$

$$\Delta\mathbf{W}^{R,i} = \mathbf{L}(\mathbf{V}^i) \cdot \Delta\mathbf{V}^{R,i}, \tag{5.13}$$

$$\Delta\mathbf{W}^{C,i} = \mathbf{L}(\mathbf{V}^i) \cdot \Delta\mathbf{V}^{C,i}. \tag{5.14}$$

The BAL approach does this the other way around: the primitive variables are transformed *before* the differences are found. We then apply a monotonicity constraint to ensure that the reconstruction is total variation diminishing. PLM and PPM use the following constraint:

$$\Delta W^i_{\alpha,m} = \mathrm{sgn}(\Delta W^{C,i}_\alpha)\min(2|\Delta W^{L,i}_\alpha|, 2|\Delta W^{R,i}_\alpha|, |\Delta W^{C,i}_\alpha|), \tag{5.15}$$

where $\alpha \in \{1...7\}$ are the elements of each vector, while BAL uses:

$$\Delta W^i_{\alpha,m} = \mathrm{sgn}(\Delta W^{R,i}_\alpha)\min(\Psi|\Delta W^{L,i}_\alpha|, \Psi|\Delta W^{R,i}_\alpha|, |\Delta W^{C,i}_\alpha|), \tag{5.16}$$

if $(\Delta W^{R,i}_\alpha)(\Delta W^{L,i}_\alpha) > 0$, and $\Delta W^i_{\alpha,m} = 0$ otherwise. The factor Ψ is set to 2 for genuinely non-linear characteristic fields (the fast and slow magnetohydrodynamic waves), and 1 for the linearly degenerate characteristic fields (the two Alfvén waves and the entropy wave). This difference is then projected back into the primitive variables, to get the required slope:

$$\Delta\mathbf{V}^i = \sum_\alpha \Delta W^i_{\alpha,m} R_\alpha(\mathbf{V}^i). \tag{5.17}$$

This is all that is required for PLM. However, BAL and PPM have additional steps:

- BAL uses a steepening algorithm on the linearly degenerate characteristic fields, to counteract the numerical 'smearing' of these fields that tends to occur with characteristic interpolation. This is described in section 2.2 of Balsara (1998), specifically their eqs. (2.5)-(2.12). This takes the form of a correction term $\delta\mathbf{V}^i$, such that $\Delta\mathbf{V}^i \rightarrow \Delta\mathbf{V}^i + \delta\mathbf{V}^i$.
- PPM uses further monotonicity constraints, given by eqs. (47)-(53) of Stone *et al.* (2008).

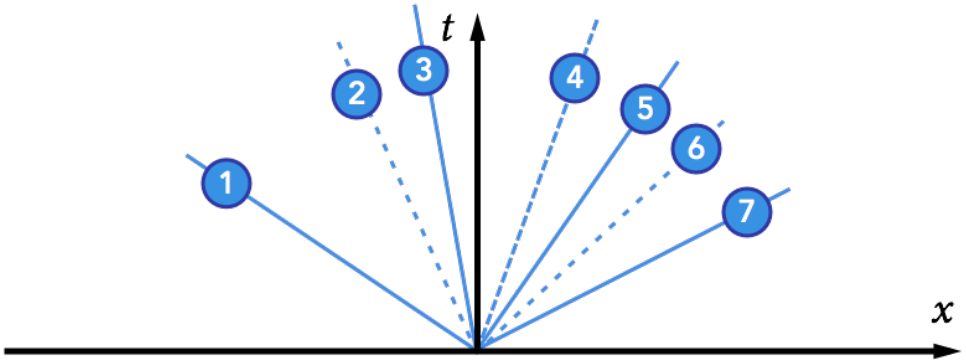


Figure 5.1: A typical Riemann solution for the MHD equations, with seven waves. Waves 1 and 7 are the fast magnetosonic waves, 2 and 6 are the Alfvén waves, 3 and 5 are the slow magnetosonic waves and 4 is the entropy wave/contact discontinuity.

(c) Time Integral and Characteristic Tracing

Finally, we complete the two additional steps described by Colella (1990). The first step is to determine the two reference states, $\hat{\mathbf{V}}_L^{i*}$ and $\hat{\mathbf{V}}_R^{i*}$. For BAL and PLM:

$$\tilde{\mathbf{V}}_L^{i*} = \mathbf{V}^i + \left[\frac{1}{2} - \max(\lambda_7^i, 0)\frac{\Delta t}{2\Delta x}\right]\Delta\mathbf{V}^i, \tag{5.18}$$

$$\tilde{\mathbf{V}}_R^{i*} = \mathbf{V}^{i+1} - \left[\frac{1}{2} + \min(\lambda_1^{i+1}, 0)\frac{\Delta t}{2\Delta x}\right]\Delta\mathbf{V}^{i+1}, \tag{5.19}$$

where λ_1^{i+1} and λ_7^{i+1} are the largest and smallest eigenvalue respectively. The second step is then to correct this integral using characteristic tracing:

$$\mathbf{V}_{n*,L}^{i*} = \hat{\mathbf{V}}_L^{i*} + \frac{\Delta t}{2\Delta x}\sum_{\alpha,\lambda_\alpha^i>0}(\lambda_7^i - \lambda_\alpha^i)\Delta W_\alpha^i R_\alpha(\mathbf{V}^i), \tag{5.20}$$

$$\mathbf{V}_{n*,R}^{i*} = \hat{\mathbf{V}}_R^{i*} + \frac{\Delta t}{2\Delta x}\sum_{\alpha,\lambda_\alpha^{i+1}<0}(\lambda_1^{i+1} - \lambda_\alpha^{i+1})\Delta W_\alpha^{i+1} R_\alpha(\mathbf{V}^{i+1}). \tag{5.21}$$

And so we have obtained the required left and right input states for the Riemann solver, $\mathbf{V}_{n*,L}^{i*}$ and $\mathbf{V}_{n*,R}^{i*}$. The higher-order PPM algorithm requires more elaborate versions of eqs. (5.18)-(5.21); these are given in eqs. (55)-(60) of Stone *et al.* (2008).

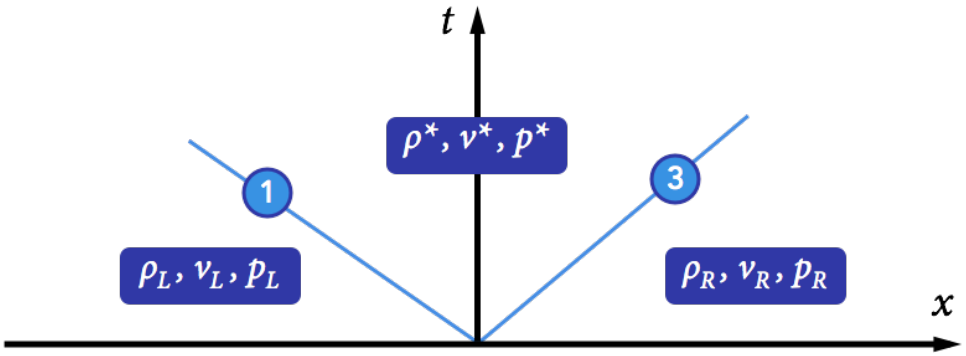


Figure 5.2: The HLL solver's approximation to the solution of the Riemann problem for hydrodynamics. There are only two waves, with wavespeeds approximated by S_L and S_R. The single internal starred state is then a function of these two wavespeeds and the left and right input states.

5.3 Upgrading the Solve Stage

We now turn our attention to the Riemann solver algorithm. This upgrade is not as simple as the reconstruct step, because we have to deal with *seven* possible waves instead of three, as shown in Fig. 5.1: two fast magneto-acoustic waves (1 and 7), two Alfvén waves (2 and 6), two slow magneto-acoustic waves (3 and 5) and the entropy wave (4).

Exact Riemann solvers have been developed for MHD, but these are algorithmically complex and computationally expensive. Hence we use an *approximate* Riemann solver, as discussed in Section 2.4.4. These are simpler and cheaper, and return only the minimum information required for the RSA strategy: the intercell fluxes. In this chapter, we use the HLLD Riemann solver of Miyoshi and Kusano (2005), which is part of the HLL family of solvers.

5.3.1 The HLL and HLLC Solvers

The original HLL Riemann solver was designed for the Euler equations, and is named for its developers, Harten *et al.* (1983). Recall from Fig. 2.3 that the Riemann solution for hydrodynamics involves three waves: the two outer waves can be shock or rarefaction waves, while the central wave is a contact discontinuity. The HLL model simplifies this solution to just two simple outer waves, with wavespeeds S_L and S_R, as shown in Fig. 5.2. Assuming we have some method for calculating the two wavespeeds, the internal starred state is given by:

$$\mathbf{U}^* = \frac{S_R\mathbf{U}_R - S_L\mathbf{U}_L - (\mathbf{F}_R - \mathbf{F}_L)}{S_R - S_L}, \tag{5.22}$$

where we have dropped the position and time indices for clarity. The corresponding flux is:

$$\mathbf{F}^* = \frac{S_R\mathbf{F}_L - S_L\mathbf{F}_R + S_LS_R(\mathbf{U}_R - \mathbf{U}_L)}{S_R - S_L}. \tag{5.23}$$

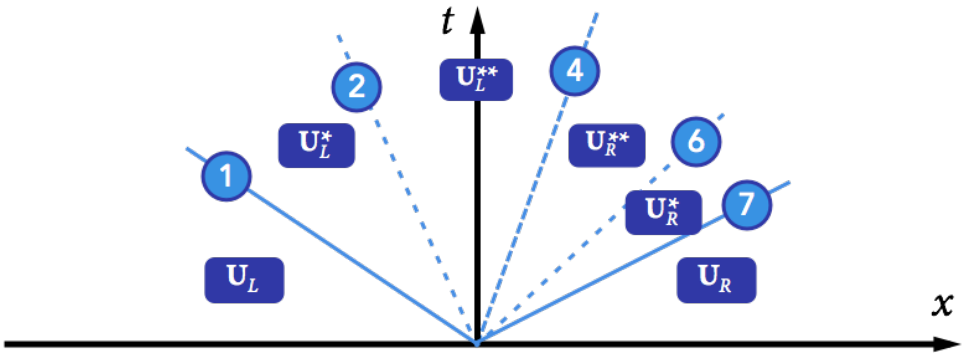


Figure 5.3: The HLLD solver's approximation to the solution of the MHD Riemann problem. There are five waves, with wavespeeds S_L, S_L^*, S_M, S_R^* and S_L. This leads to four internal states: $\mathbf{U}_*^L$, $\mathbf{U}_{**}^L$, $\mathbf{U}_{**}^R$, and $\mathbf{U}_*^R$, which can be determined using the wavespeeds and the left and right input states.

The intercell flux we require is then given by:

$$\mathbf{F}_{\text{HLL}} = \begin{cases} \mathbf{F}_L & S_L \geq 0, \\ \mathbf{F}^* & S_L \leq 0 \leq S_R, \\ \mathbf{F}_R & S_R \leq 0, \end{cases} \tag{5.24}$$

where $\mathbf{F}_{L/R}$ can be calculated using the flux vectors given by eq. (2.24). It is clear that the strength of this method depends on the estimates of the wavespeeds S_L and S_R.

The obvious disadvantage of this approach is that it ignores the central contact discontinuity, which can lead to diffusive results. Toro *et al.* (1994) attempted to solve this problem with the development of the HLLC solver. This approximates the contact discontinuity by introducing a third wave, which is defined as having wavespeed $S_M = v_x$. This results in two internal starred states, $\mathbf{U}_L^*$ – given by their eqs. (22)-(24) – and $\mathbf{U}_R^*$ – given by their eqs. (26)-(28). The corresponding flux vectors, $\mathbf{F}_L^*$ and $\mathbf{F}_R^*$ are given by their eqs. (18) and (19) respectively. The required intercell flux is then determined using:

$$\mathbf{F}_{\text{HLLC}} = \begin{cases} \mathbf{F}_L & S_L \geq 0, \\ \mathbf{F}_L^* & S_L \leq 0 \leq S_M, \\ \mathbf{F}_R^* & S_M \leq 0 \leq S_R, \\ \mathbf{F}_R & S_R \leq 0. \end{cases} \tag{5.25}$$

5.3.2 The HLLD Solver

Attempts have been made to extend the HLLC solver to MHD (e.g. Li, 2005). However, these maintain the three-wave model, which is a poor approximation of the seven-wave structure of MHD. This shortcoming led to the development of the HLLD solver of Miyoshi and Kusano (2005), which extends the HLLC solver to *five* waves – ignoring the two slow magneto-acoustic

waves – and hence has four internal states: $\mathbf{U}_L^*$, $\mathbf{U}_L^{**}$, $\mathbf{U}_R^{**}$ and $\mathbf{U}_R^*$. This is depicted in Fig. 5.3. As before, we start by estimating the two outer wavespeeds, S_L and S_R. Two estimates have been implemented in MAGNETO; the first is eq. (67) of Miyoshi and Kusano (2005):

$$S_L = \min(v_L, v_R) - \max(c_{f,L}, c_{f,R}), \qquad S_R = \max(v_L, v_R) + \max(c_{f,L}, c_{f,R}), \tag{5.26}$$

where $c_{f,L/R}$ is the speed of the fast magneto-acoustic wave, which is calculated when we determine the eigensystem. The second estimate is eq. (28) of Janhunen (2000):

$$S_L = \min(v_L, v_R) - \hat{c}_f, \qquad S_R = \max(v_L, v_R) + \hat{c}_f, \tag{5.27}$$

where $\hat{c}_f$ is the maximum possible speed of the fast magneto-acoustic wave (given the left and right input states), and is calculated using eq. (27) of their paper. The choice of wavespeed can be changed in MAGNETO in `start.txt`, but we have generally found the Miyoshi and Kusano (2005) choice to be more stable and robust during testing.

The speed of the entropy wave, S_M, is now approximated as:

$$S_M = \frac{(S_R - v_R)\rho_R v_R - (S_L - v_L)\rho_L v_L - p_{T,R} + p_{T,L}}{(S_R - v_R)\rho_R - (S_L - v_L)\rho_L}. \tag{5.28}$$

Once we know S_L, S_R and S_M, we can determine the two single starred states – $\mathbf{U}_L^*$ and $\mathbf{U}_R^*$ – using eqs. (39)-(41), (43)-(48) of their paper. We then use the values of ρ_L^* and ρ_R^* to estimate the speed of the two Alfvén waves, which we call S_L^* and S_R^*:

$$S_L^* = S_M - \frac{|(Bx)|}{\sqrt{\rho_L^*}}, \qquad S_R^* = S_M + \frac{|(Bx)|}{\sqrt{\rho_R^*}}. \tag{5.29}$$

Finally, we use these to determine the two double starred states, $\mathbf{U}_L^{**}$ and $\mathbf{U}_R^{**}$, using eqs. (49), (50), (59)-(63) of their paper. Once we have all the internal states, we can determine the corresponding fluxes – $\mathbf{F}_L^*$, $\mathbf{F}_L^{**}$, $\mathbf{F}_R^{**}$ and $\mathbf{F}_R^*$ – by substituting the appropriate variables into the MHD flux vector given in eq. (5.1). The final intercell flux is then given by:

$$\mathbf{F}_{\mathrm{HLLD}}^{i*} = \begin{cases} \mathbf{F}_L & S_L \geq 0, \\ \mathbf{F}_L^* & S_L \leq 0 \leq S_L^*, \\ \mathbf{F}_L^{**} & S_L^* \leq 0 \leq S_M, \\ \mathbf{F}_R^{**} & S_M \leq 0 \leq S_R^*, \\ \mathbf{F}_R^* & S_R^* \leq 0 \leq S_R, \\ \mathbf{F}_R & S_R \leq 0. \end{cases} \tag{5.30}$$

This solver is implemented in subroutine `[SOL02]` of MAGNETO.

	SOD		**BW**		**RJ2A**		**RJ4D**	
	L	R	L	R	L	R	L	R
ρ	1	0.125	1	0.125	1.08	1	1	0.3
P	1	0.1	1	0.1	0.95	1	1	0.2
v_x	0	0	0	0	1.2	0	0	0
v_y	0	0	0	0	0.01	0	0	0
v_z	0	0	0	0	0.5	0	0	1
B_y	0	0	1	-1	$3.6/\sqrt{4\pi}$	$4/\sqrt{4\pi}$	0	1
B_z	0	0	0	0	$2/\sqrt{4\pi}$	$2/\sqrt{4\pi}$	0	0
B_x	0		0.75		$2/\sqrt{4\pi}$		0.7	
γ	1.4		2		5/3		5/3	
t	0.25		0.08		0.2		0.16	

Table 5.1: Initial conditions for all four shock tube tests.

5.4 1D MHD Tests

We conclude this chapter with some 1D tests, to show that the algorithms described in this chapter have been implemented correctly. We will also use these results to determine which reconstruction algorithm is best for subsequent investigations.

We run one non-magnetic shock tube test – the well-known Sod test – to show that the new algorithms can handle pure hydrodynamics in the absence of a magnetic field. This is essential, as a hydrodynamics engine that has been upgraded to magnetohydrodynamics should be able to handle the limit $\mathbf{B} \to 0$. For example: in our initial magnetised merger tests, we will only initialise a magnetic field within the neutron star. Hence, at early times, most of the domain will be governed by pure hydrodynamics. We also run three magnetic shock tube tests: one test of Brio and Wu (1988) (the Sod test with a magnetic field added), and two tests from Ryu and Jones (1995). We choose these tests so that we can compare our results with those of Stone *et al.* (2008). We refer to them as SOD, BW, RJ2A and RJ4D respectively (with the latter two labels following the convention of the Stone *et al.* paper).

The initial conditions are given in Table 5.1. We run each test with all four reconstruction algorithms, and at four resolutions: $n_x = 128$, 256, 512 and 1024 cells. All graphed results are from the $n_x = 256$ case. We compare these with a high-resolution reference result obtained using 4096 cells, where the error is estimated using the standard L_1 norm on the density:

$$\epsilon_{L_1} = \sum_i |\rho^i - \rho^i_{\text{ref}}| \Delta x. \tag{5.31}$$

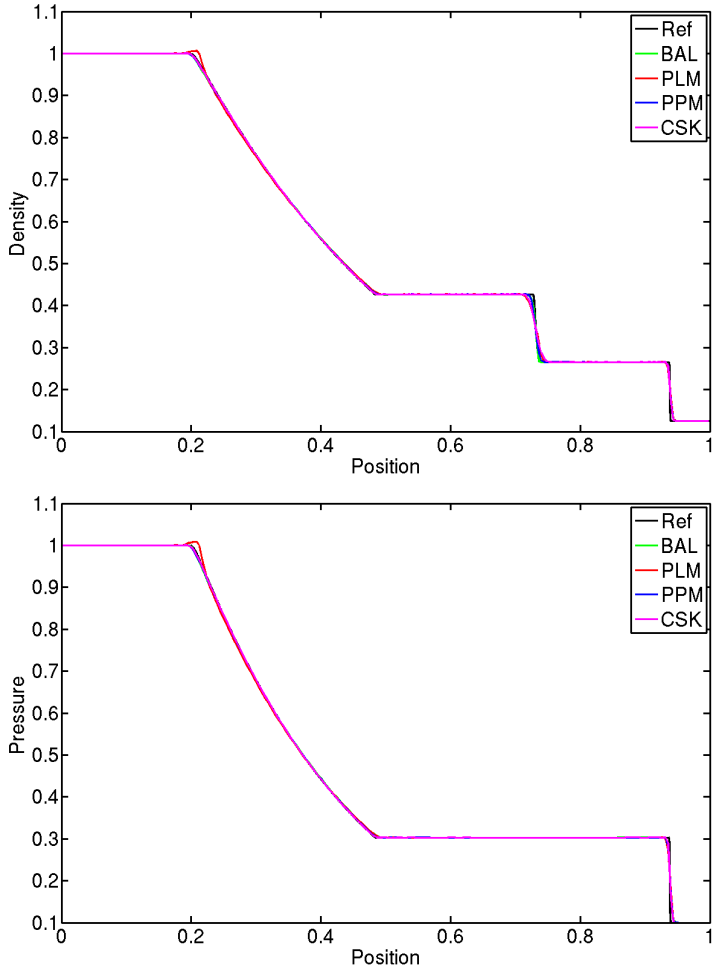


Figure 5.4: Density (top) and pressure (bottom) for the SOD test at $t = 0.25$.

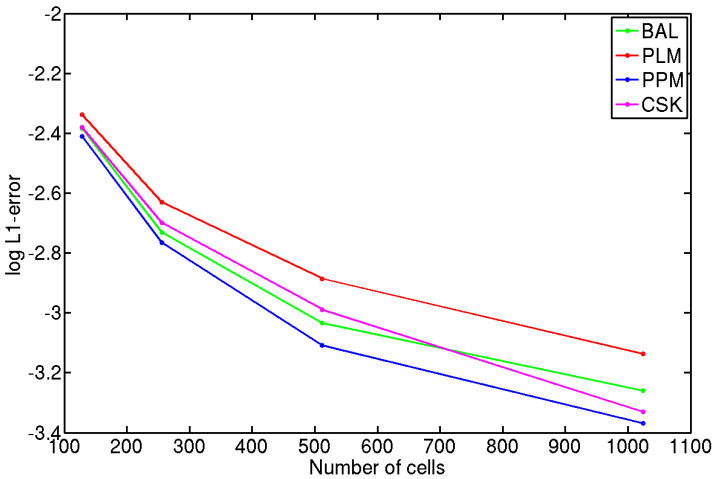


Figure 5.5: L1-error for the SOD test.

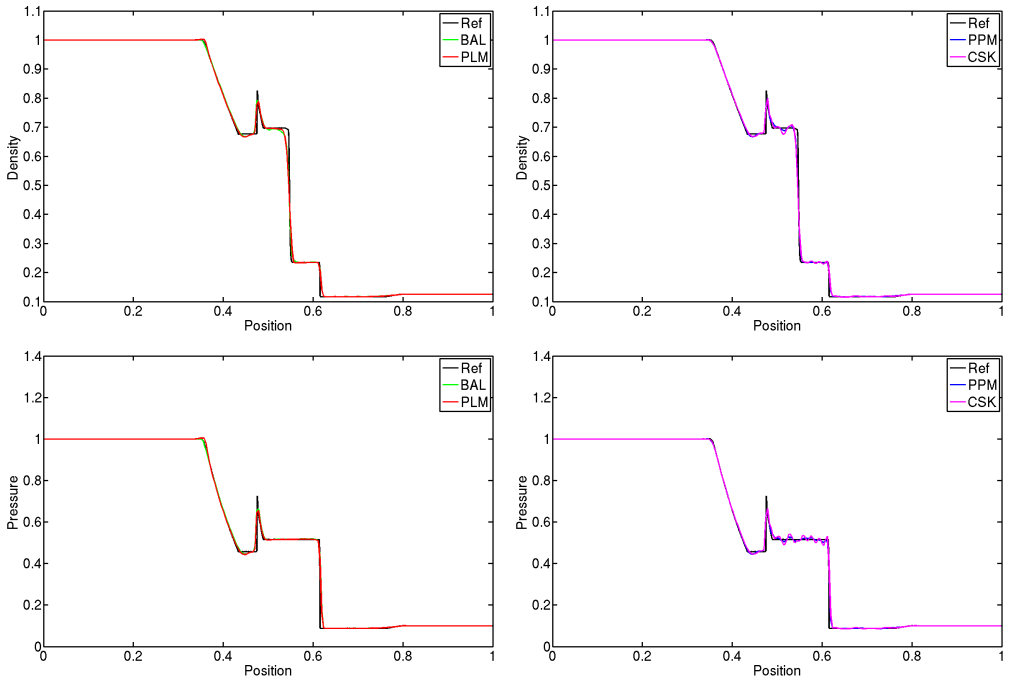


Figure 5.6: Density (top row) and pressure (bottom row) for the BW test at $t = 0.08$.

5.4.1 SOD: Sod Shock Tube Test

The density and pressure results are given in Fig. 5.4, while the error is given in Fig. 5.5. These can be compared with e.g. fig. 8 of Stone *et al.* (2008). This is a classic test for pure hydrodynamics, which produces all three characteristic features: in the density plot, we can see a rarefaction wave (left), a contact discontinuity (centre) and a shock (right). All reconstruction algorithms test this straightforward test with ease, although there is some softening of the contact discontinuity and shock. Additionally, the PLM algorithm produces some oscillations around the head of the rarefaction wave.

5.4.2 BW: Sod Shock Tube Test with a Magnetic Field

The density and pressure results are given in Fig. 5.6, the y-component of the magnetic field and the velocity are given in Fig. 5.7, and the error is given in Fig. 5.8. These can be compared with fig. 2 of Brio and Wu (1988) or fig. 13 of Stone *et al.* (2008). There are five characteristic features visible in the density plot, from left to right: a fast rarefaction wave ($x \sim 0.4$), a slow compound wave (the peak at $x \sim 0.48$), a contact discontinuity ($x \sim 0.54$), a slow shock ($x \sim 0.61$) and another fast rarefaction wave ($x \sim 0.77$) (Brio and Wu, 1988). Note that the slow shock appears, even though it is not explicitly included in the HLLD Riemann solver.

All reconstructions struggle with the peak of the slow compound wave and the tail of the rarefaction wave to the left. The BAL and PLM results are otherwise good, although PLM still has some difficulty with the head of the rarefaction wave. Both PPM and CSK suffer badly from oscillations, especially around the three right-most waves. Stone *et al.*

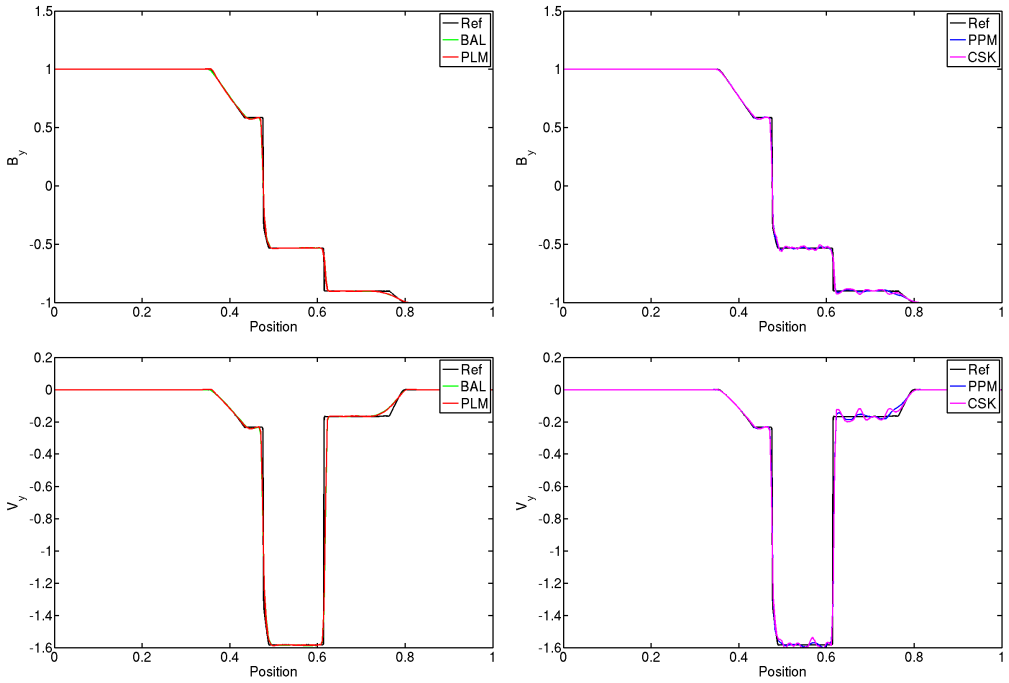


Figure 5.7: y-component of the B-field (top row) and velocity (bottom row) for the BW test at $t = 0.08$.

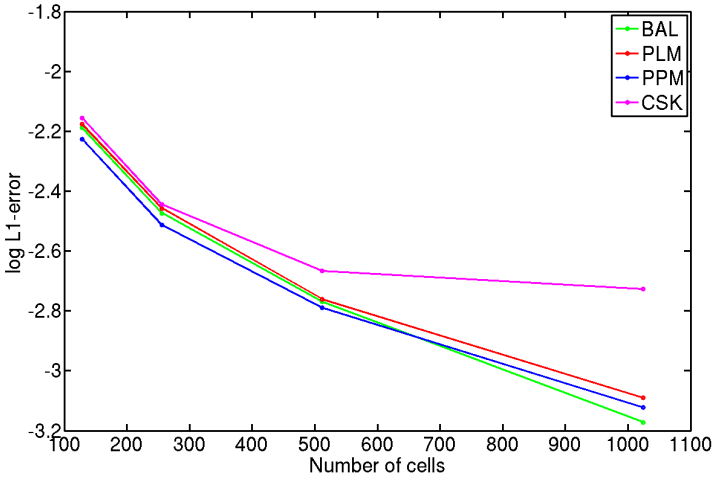


Figure 5.8: L1-error for the BW test.

(2008) finds these oscillations present in the velocity for PPM, but we find them present in all graphed quantities. However, they are considerably worse in the CSK case, suggesting that characteristic interpolation can help reduce oscillations. Interestingly, PPM still results in the lowest overall error (with the exception of BAL at $n_x = 1024$).

An Aside on Entropy and Evolutionary Conditions

Not all discontinuities that arise from hyperbolic conservation laws are valid. For example, some can be excluded for physical reasons, such as the *entropy condition* implied by the second law of thermodynamics. An example of an entropy-violating discontinuity is the 'rarefaction shock' in gas dynamics: a discontinuity which moves towards a region of high density – leaving a rarefied, lower density region in its wake – hence *decreasing* the entropy and violating the second law. However, discontinuties can also be excluded for mathematical reasons – such as uniqueness or stability – and these are often described as *evolutionary conditions*.

Falle and Komissarov (2001) note that intermediate shocks in MHD satisfy the entropy condition but are not evolutionary. Additionally, there are some shocks that are not evolutionary in full, three-dimensional MHD but *are* evolutionary in 'planar' MHD (investigated in this chapter), and vice-versa. The compound wave in the BW test – a slow shock combined with a rarefaction, which was first observed in numerical simulations – is one such wave, and is therefore not valid. In fact, Falle and Komissarov (2001) find that this wave disappears when they use a numerical scheme that disallows non-evolutionary shocks, as shown in figure 8 of their paper. Hence, we must evaluate these results with caution: we use them only as a first check for the validity of our algorithms and implementation against previous papers (which also include this wave). However, we also need to perform further checks in full, three-dimensional MHD – and these are presented in the next chapter.

5.4.3 RJ2A: First Test of Ryu and Jones

The density and pressure results are given in Fig. 5.9, the y-component of the magnetic field and the velocity are given in Fig. 5.10, while the z-components are given in Fig. 5.11. The error is given in Fig. 5.12. These can be compared with fig. 2a of Ryu and Jones (1995) or fig. 14 of Stone *et al.* (2008). This test is useful, because we see evidence of all seven MHD waves across the graphed quantities: the contact discontinuity at $x \sim 0.61$, and then a slow shock, rotational discontinuity and fast shock going outwards in either direction (Ryu and Jones, 1995). Note that a rotational discontinuity is one where the direction of the magnetic field changes arbitrarily, while the magnitude remains the same.

All reconstructions handle this test moderately well, although struggle with the proximity of the slow shocks and the rotational discontinuities in the velocity and magnetic field components. Oscillations are still present for the PPM and CSK results, but these are not as bad as for the BW test. They are also better than BAL and PLM at resolving sharp features, e.g. the right-hand slow shock in the density ($x \sim 0.68$). We note that BAL is now the worst-performing of the characteristic reconstructions at the highest resolution.

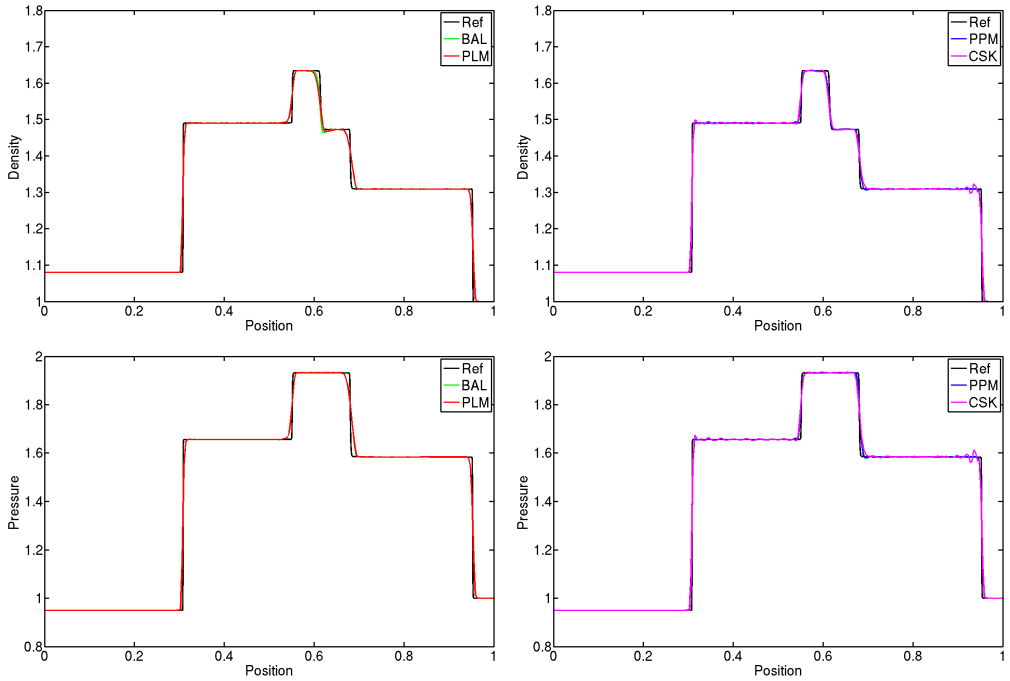


Figure 5.9: Density (top row) and pressure (bottom row) for the RJ2A test at $t = 0.2$.

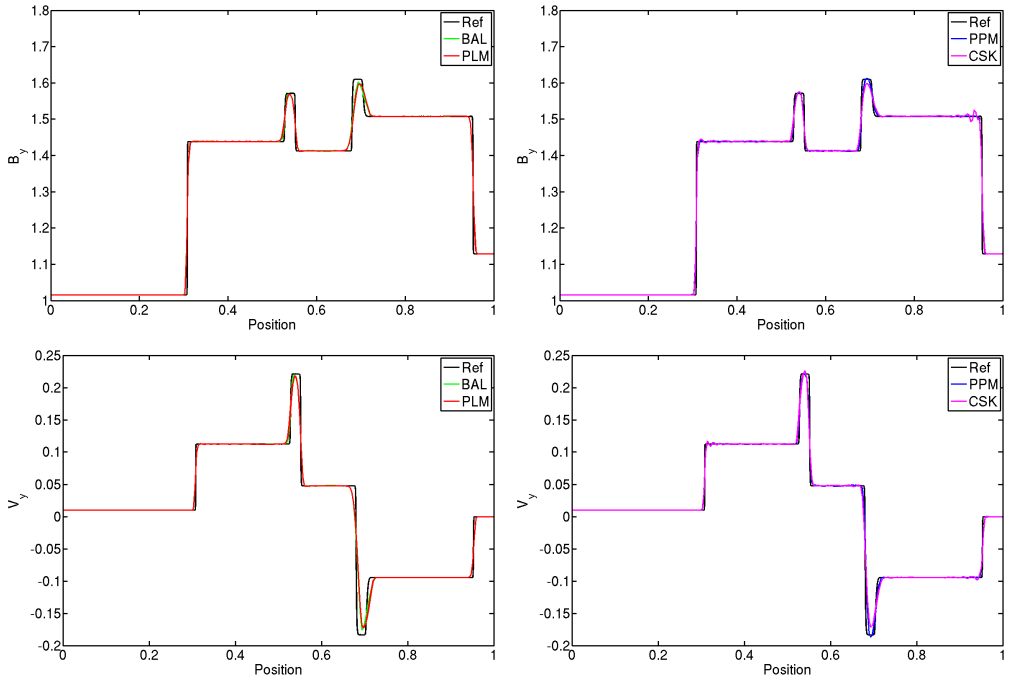


Figure 5.10: y-component of the B-field (top row) and velocity (bottom row) for the RJ2A test at $t = 0.2$.

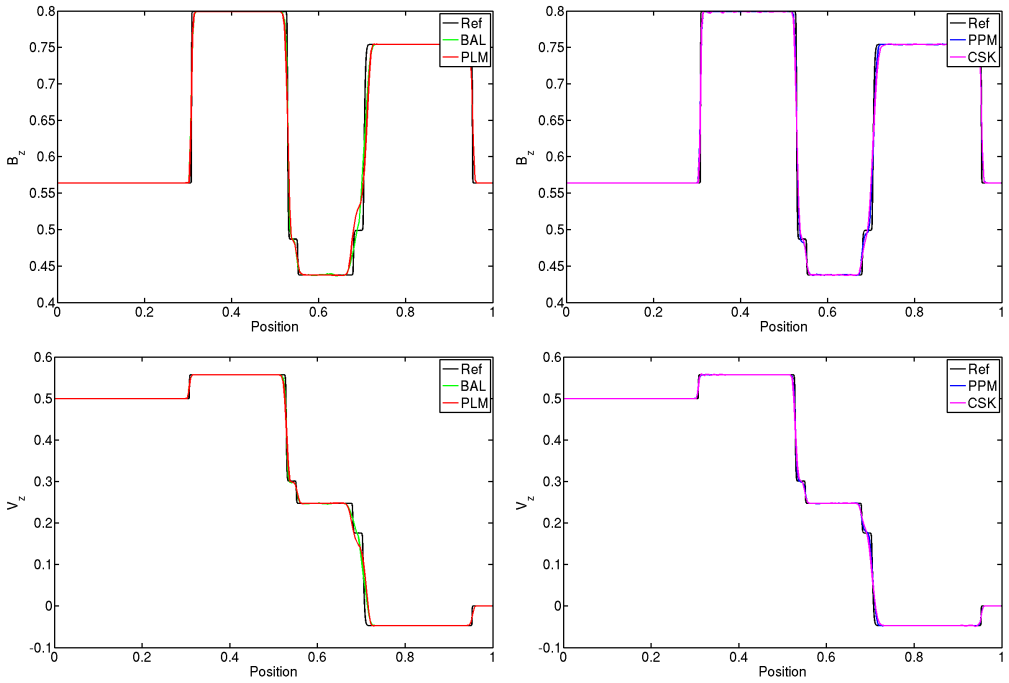


Figure 5.11: z-component of the B-field (top row) and velocity (bottom row) for the RJ2A test at $t = 0.2$.

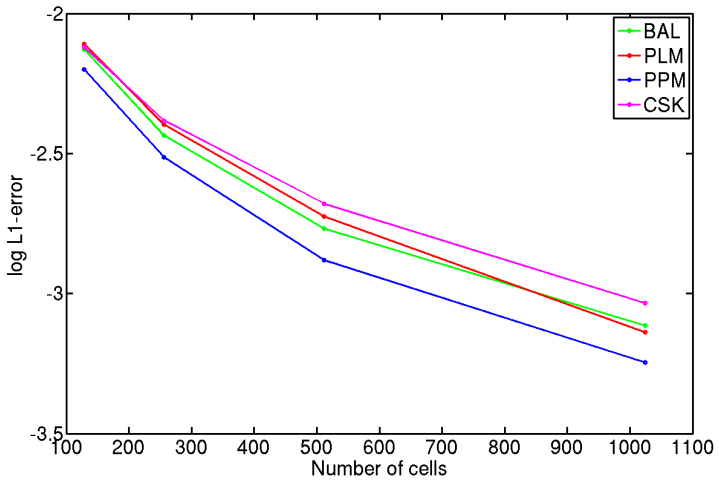


Figure 5.12: L1-error for the RJ2A test.

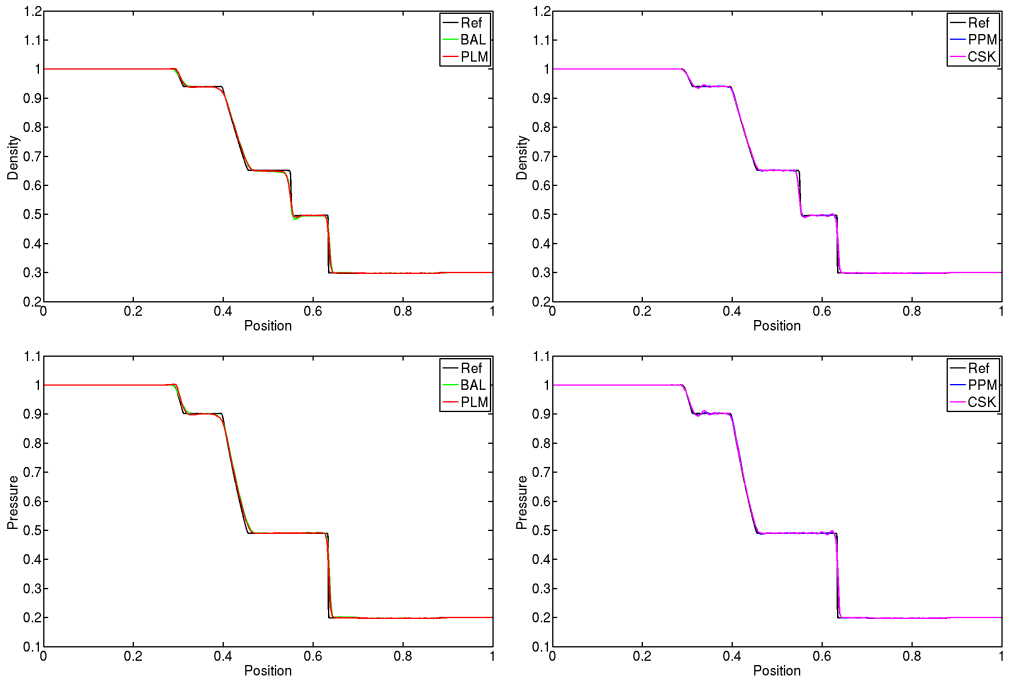


Figure 5.13: Density (top) and pressure (bottom) for the RJ4D test at $t = 0.2$.

5.4.4 RJ4D: Second Test of Ryu and Jones

The density and pressure results are given in Fig. 5.13, the y-component of the magnetic field and the velocity are given in Fig. 5.14, and the z-components are given in Fig. 5.15. The error is given in Fig. 5.16. These can be compared with fig. 4d of Ryu and Jones (1995) or fig. 15 of Stone *et al.* (2008). This test is of interest because – as pointed out by Stone *et al.* (2008) – it includes a so-called 'switch-on slow rarefaction', where the z-component of the magnetic field 'turns on' behind the rarefaction from an initial state of zero.

PPM and CSK handle this test well, although (like all the reconstructions) struggle with the fast shock heading towards the right ($x \sim 0.71$). This is most visible in the magnetic field and velocity components. BAL and PLM now perform the worst, creating notable new local extrema around the slow shock and the slow rarefaction wave ($x \sim 0.42$ and $x \sim 0.63$ respectively), in both B_z and v_z. Consequently, this is the first test where CSK produces consistently smaller error than BAL and PLM across all resolutions.

5.5 Conclusions

In this chapter, we have implemented four reconstruction algorithms in a test code, Magneto, and presented 1D tests to compare them. Based on these results, we find that the PPM reconstruction of Stone *et al.* (2008) will be the best choice for our simulations of BH-NS mergers, as it consistently produces the lowest L1-error. The only exception is the BW test at $n_x = 1024$, when the second-order reconstruction of Balsara (1998) (BAL) performs better. However, we note that *all* algorithms produce unphysical oscillations in at least one test.

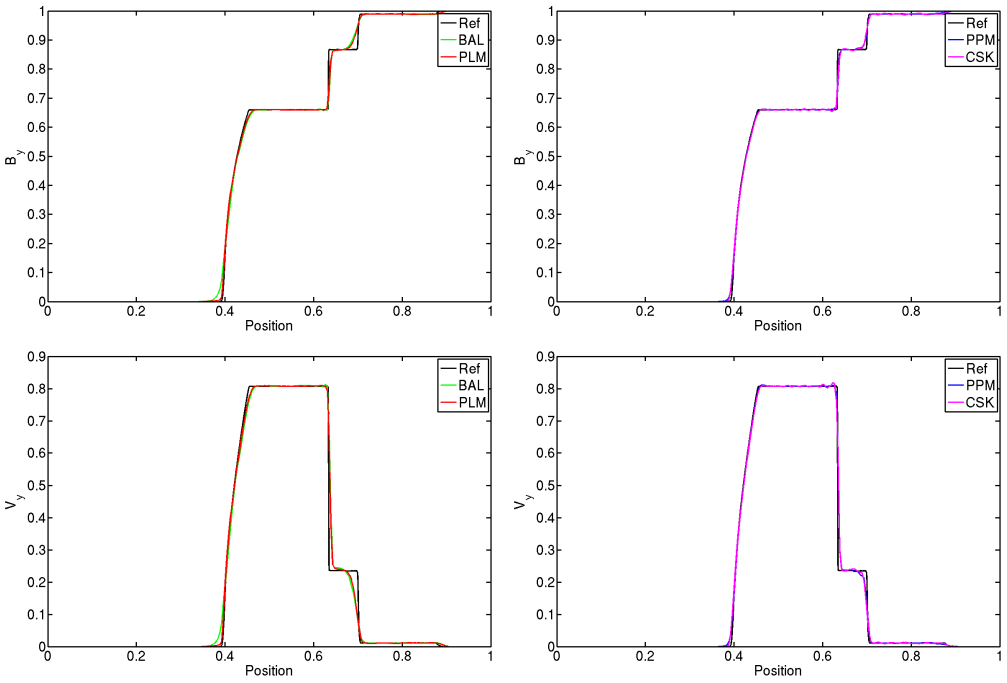


Figure 5.14: y-component of the B-field (top) and velocity (bottom) for the RJ4D test at $t = 0.2$.

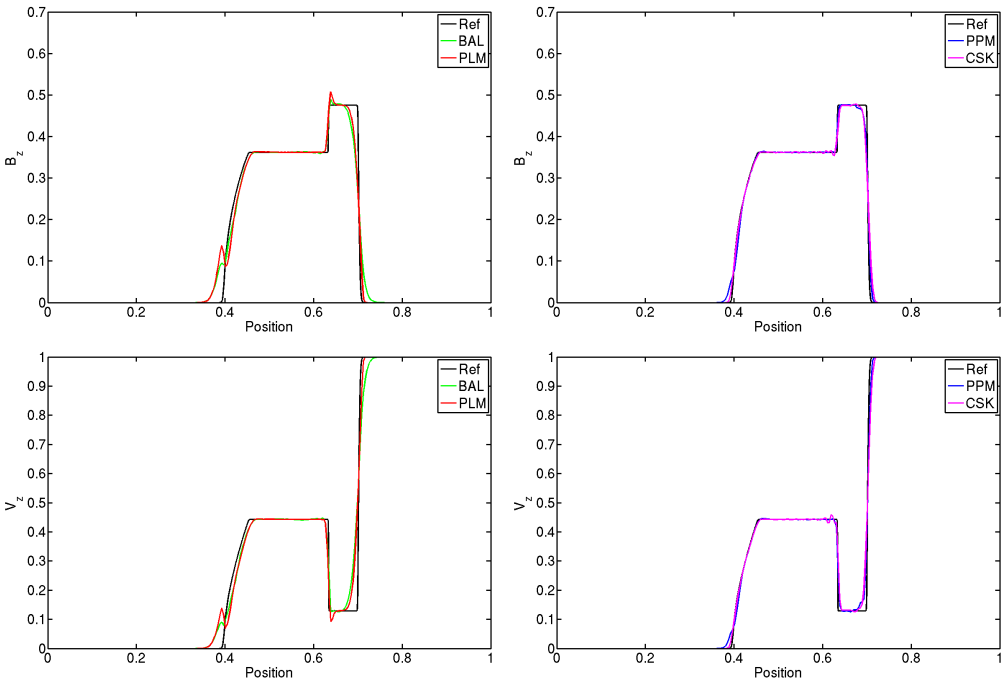


Figure 5.15: z-component of the B-field (top) and velocity (bottom) for the RJ4D test at $t = 0.2$.

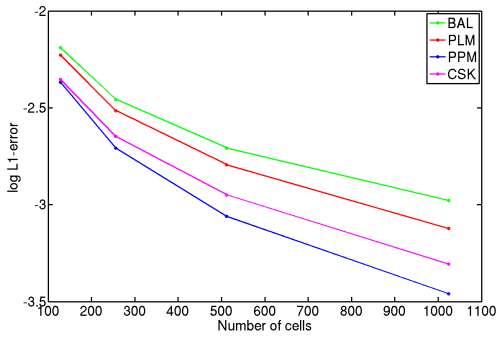


Figure 5.16: L1-error for the RJ4D test.

Balsara (1998) notes that both primitive and characteristic interpolation work well for the Euler equations. This can be seen in our SOD results, where there is no clear correlation between the error and type of interpolation. For the MHD results, however, we find that characteristic interpolation does tend to result in lower error. There is one exception to this, the RJ4D test, where the reconstruction of Colella and Sekora (2008) (CSK) performs consistently better than both BAL and the reconstruction of Stone *et al.* (2008) (PLM). However, this could be because both BAL and PLM are second-order. CSK still does not perform better than PPM, which is third-order *and* uses characteristic interpolation.

We also implemented the HLLD approximate Riemann solver of Miyoshi and Kusano (2005). We did not test this against other Riemann solvers, but our results suggest that it is mostly working correctly. However, there are a few cases where errors were consistent across all reconstruction algorithms. It is possible that the HLLD solver caused these, and hence further study is required to determine whether this is the case.

Chapter 6
Modelling Magnetic Fields (2D/3D)

Modelling magnetic fields in more than one dimension is complicated by the *divergence-free condition*, $\nabla \cdot \mathbf{B} = 0$. Although many algorithms exist for maintaining this condition, they are often difficult to implement in mature astrophysical codes, and remove several advantages of the DS-RSA strategy. In this chapter, we show we can minimise these issues by presenting a variation of the existing Balsara-Spicer-Tóth algorithm (Balsara and Spicer, 1999; Tóth, 2000). We also present several 2D test problems to demonstrate its accuracy and robustness.

Note that this chapter is based on a paper that has been published in the *Monthly Notices of the Royal Astronomical Society* (Sriskantha and Ruffert, 2013).

6.1 Introduction

As we have seen in the previous chapter, extending a 1D hydrodynamics code to magneto-hydrodynamics is mostly straightforward. However, matters are complicated in two or more dimensions, because of the need to maintain the divergence-free condition[1]:

$$\nabla \cdot \mathbf{B} = 0. \tag{6.1}$$

It has been shown that ignoring this condition in numerical simulations can result in unwanted effects, such as the creation of unphysical forces parallel to the magnetic field, and the loss of conversation of momentum and energy (Balsara and Spicer, 1999).

There are three main families of algorithms for maintaining this condition, which are used with the reconstruct-solve-average (RSA) strategy described in Chapter 2 (Tóth, 2000):

- **8-wave formulation:** an eighth equation is added to the MHD equations (and hence an eighth wave to the solution). This is an advection equation for $\nabla \cdot \mathbf{B}$, which is designed to prevent excessive non-zero divergence from building up anywhere in the system.

[1] This condition is equally important in 1D, but is automatically satisfied if the parallel component of the magnetic field is kept uniform in space and constant in time. This was the case for all the 1D tests we presented.

- **Constrained transport:** the magnetic field quantities are discretised in such a way that $\nabla \cdot \mathbf{B}$ does not change in time analytically (to machine error). We then choose initial conditions such that $\nabla \cdot \mathbf{B} = 0$, which is always the case for physical models.
- **Hodge projection:** after the solution for the magnetic field is obtained from the RSA scheme, it is projected onto the subspace of divergence-free solutions. This projected (and hence 'cleaned') solution is then used in the next timestep.

Here, we focus on the constrained transport (CT) approach. Many CT algorithms have been developed, such as: Dai and Woodward (1998); Balsara and Spicer (1999); Tóth (2000); Londrillo and del Zanna (2000). However, these are often dimensionally unsplit and/or require a 'staggered' grid: the fluid quantities are stored as the usual cell averages, but the magnetic field quantities are stored as *face* averages. This can complicate their implementation in mature astrophysical codes – such as Charybdis – which are dimensionally split and use some form of adaptive grids to model phenomena on different length scales.

Not all CT algorithms require a staggered grid, such as that of Tóth (2000), which is modified from the algorithm of Balsara and Spicer (1999). However, even this involves considerable restructuring of the underlying hydrodynamics engine, and – as described below – removes several advantages of the dimensionally-split RSA (DS-RSA) approach. In this chapter, we derive a variation of the Balsara-Spicer-Tóth algorithm which minimises this restructuring and restores many of the advantages of a pure DS-RSA code. By doing so, we aim to considerably simplify the development required to add magnetic fields to Charybdis.

6.2 The Balsara-Spicer-Tóth Algorithm

First, we summarise the derivation of this algorithm. The fluid quantities are evolved using the standard DS-RSA strategy, using the MHD reconstruction and Riemann solver algorithms described in Chapter 5. The magnetic field quantities are evolved using constrained transport.

6.2.1 Constrained Transport

Constrained transport was first developed by Evans and Hawley (1988), and uses Faraday's law to determine how the magnetic field $\mathbf{B}$ changes in time:

$$\frac{\partial}{\partial t}\mathbf{B} = -\nabla \times \mathbf{E}, \tag{6.2}$$

where $\mathbf{E}$ is the electric field. If we take the divergence of both sides of this equation, we get:

$$\frac{\partial}{\partial t}\nabla \cdot \mathbf{B} = -\nabla \cdot (\nabla \times \mathbf{E}) = 0, \tag{6.3}$$

where the second equality is true because $\nabla \cdot (\nabla \times \mathbf{F}) = 0$ for any vector $\mathbf{F}$. This means that if we evolve the magnetic field using eq. (6.2), then $\nabla \cdot \mathbf{B}$ will not change in time analytically. We can therefore maintain the divergence-free condition by simply choosing initial conditions such that $\nabla \cdot \mathbf{B} = 0$, which will be true for any physical models. To implement eq. (6.2)

numerically, we start by considering the x-component of the magnetic field averaged on the face $(i*)$, which we call $(bx)^{i*}$. If we define $(i*)$ as having area ΔA, then it follows that:

$$\frac{\partial}{\partial t}(bx)^{i*} = -\frac{1}{\Delta A}\int_A \nabla \times \mathbf{E} \cdot d\mathbf{A}. \tag{6.4}$$

Note the use of the lower case b to denote a face-averaged magnetic field quantity. (We will later use the upper-case B to distinguish the *cell*-averaged magnetic field quantities.) Using Stokes' theorem, we can rewrite eq. (6.4) as:

$$\frac{\partial}{\partial t}(bx)^{i*} = -\frac{1}{\Delta A}\int_{\partial A} \mathbf{E} \cdot d\mathbf{r}, \tag{6.5}$$

where we now integrate over the path ∂A, which is the boundary that surrounds the face $(i*)$. We can discretise this equation, because we know this boundary comprises four edges in discretised space: $(i*, j*)$, $(i*, j*-1)$, $(i*, k*)$ and $(i*, k*-1)$. Furthermore, if we integrate in time, and use the fact that $\Delta A = \Delta y \Delta z$, then $(bx)^{i*}_{n+1}$ is given by:

$$(bx)^{i*}_{n+1} = (bx)^{i*}_{n} - \frac{\Delta t}{\Delta y}\left[(Ez)^{i*,j*}_{n*} - (Ez)^{i*,j*-1}_{n*}\right] + \frac{\Delta t}{\Delta z}\left[(Ey)^{i*,k*}_{n*} - (Ey)^{i*,k*-1}_{n*}\right], \tag{6.6}$$

where $(Ez)^{i*,j*}$ is the z-component of the electric field, averaged on the *edge* $(i*, j*)$:

$$(Ez)^{i*,j*} = \frac{1}{\Delta z}\int_{(i*,j*)} \mathbf{E} \cdot d\mathbf{z}. \tag{6.7}$$

By substituting eq. (6.7) into eq. (6.6), it should be clear how eq. (6.6) follows from eq. (6.5). Using similar reasoning, we can then derive expressions for the other two (face-averaged) components of the magnetic field, $(by)^{j*}_{n+1}$ and $(bz)^{k*}_{n+1}$:

$$(by)^{j*}_{n+1} = (by)^{j*}_{n} - \frac{\Delta t}{\Delta z}\left[(Ex)^{j*,k*}_{n*} - (Ex)^{j*,k*-1}_{n*}\right] + \frac{\Delta t}{\Delta x}\left[(Ez)^{i*,j*}_{n*} - (Ez)^{i*-1,j*}_{n*}\right], \tag{6.8}$$

$$(bz)^{k*}_{n+1} = (bz)^{k*}_{n} - \frac{\Delta t}{\Delta x}\left[(Ey)^{i*,k*}_{n*} - (Ey)^{i*-1,k*}_{n*}\right] + \frac{\Delta t}{\Delta y}\left[(Ex)^{j*,k*}_{n*} - (Ex)^{j*-1,k*}_{n*}\right]. \tag{6.9}$$

6.2.2 The Electric Field

The next step is to determine the edge-averaged electric field components in eqs. (6.6), (6.8) and (6.9). Balsara and Spicer (1999) use the fact that these components appear as fluxes in the last three rows of the MHD equations:

$$\frac{\partial}{\partial t}\begin{bmatrix} \dots \\ B_x \\ B_y \\ B_z \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} \dots \\ 0 \\ -E_z \\ E_y \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} \dots \\ E_z \\ 0 \\ -E_x \end{bmatrix} + \frac{\partial}{\partial z}\begin{bmatrix} \dots \\ -E_y \\ E_x \\ 0 \end{bmatrix} = 0. \tag{6.10}$$

Recall that these are returned as intercell fluxes by the Riemann solver, which means we obtain *face*-averaged values of these components during the solve stage. We can then convert

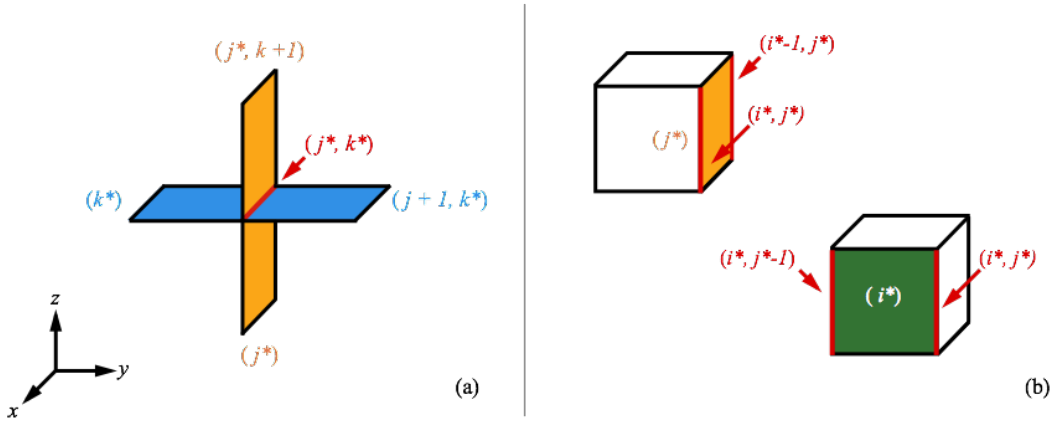


Figure 6.1: **(a) The edge-averaged electric field.** The edge $(j*,k*)$ is highlighted in red. To determine the edge average of E_x along this edge, we take the arithmetic mean of the *face* averages of E_x on the surrounding four faces. These, in turn, are returned by the Riemann solver as intercell fluxes during the solve stage. **(b) The face-interpolated electric field.** For each edge-averaged component of the magnetic field, we can define an interpolation on two faces. For example, the (Ez) component can be interpolated on the face $(j*)$ (top left) and the face $(i*)$ (bottom right). See eqs. (6.19) and (6.20).

these into the edge-averaged values we require by simple arithmetic averaging.

To make the explanation of this clearer, we re-label the last three rows of the flux vectors with the following notation (where tildes signify Riemann solver fluxes):

$$\tilde{\mathbf{F}}: \begin{bmatrix} 0 \\ (\tilde{F}7)^{i*}_{n*} \\ (\tilde{F}8)^{i*}_{n*} \end{bmatrix}, \quad \tilde{\mathbf{G}}: \begin{bmatrix} (\tilde{G}6)^{j*}_{n*} \\ 0 \\ (\tilde{G}8)^{j*}_{n*} \end{bmatrix}, \quad \tilde{\mathbf{H}}: \begin{bmatrix} (\tilde{H}6)^{k*}_{n*} \\ (\tilde{H}7)^{k*}_{n*} \\ 0 \end{bmatrix}. \tag{6.11}$$

So for example, the face-average $-(Ex)^{j*}_{n*}$ will be returned as the 8th component of the flux vector in the y-direction pass, which we label $(\tilde{G}8)^{j*}_{n*}$. Similarly, $(Ex)^{k*}_{n*}$ will be returned as 7th component of the flux vector in the z-direction pass, which we label $(\tilde{H}7)^{k*}_{n*}$.

In our grid, we know that each edge is adjacent to four faces. For example, the edge $(j*,k*)$ is adjacent to the faces $(k*)$, $(j+1,k*)$, $(j*)$ and $(j*,k+1)$, as depicted in Fig. 6.1a. So to obtain the x-component of the electric field averaged along this edge, $(Ex)^{j*,k*}_{n*}$, Balsara and Spicer simply take the arithmetic mean of the appropriate fluxes through the surrounding four faces (where minus signs here compensate for the minus signs in the flux vectors):

$$(Ex)^{j*,k*}_{n*} = \frac{1}{4}\left[(\tilde{H}7)^{k*}_{n*} + (\tilde{H}7)^{j+1,k*}_{n*} - (\tilde{G}8)^{j*}_{n*} - (\tilde{G}8)^{j*,k+1}_{n*}\right], \tag{6.12}$$

and similarly for the other components:

$$(Ey)^{i*,k*}_{n*} = \frac{1}{4}\left[(\tilde{F}8)^{i*}_{n*} + (\tilde{F}8)^{i*,k+1}_{n*} - (\tilde{H}6)^{k*}_{n*} - (\tilde{H}6)^{i+1,k*}_{n*}\right], \tag{6.13}$$

$$(Ez)_{n*}^{j*,k*} = \frac{1}{4}\left[(\tilde{G}6)_{n*}^{j*} + (\tilde{G}6)_{n*}^{i+1,j*} - (\tilde{F}7)_{n*}^{i*} - (\tilde{F}7)_{n*}^{i*,j+1}\right]. \tag{6.14}$$

This completes the description of the Balsara-Spicer algorithm.

6.2.3 The Cell-Averaged Magnetic Field

It is clear that the Balsara-Spicer algorithm requires a staggered grid: the magnetic field components we obtain from eqs. (6.6), (6.8) and (6.9) are face averages, while the fluid quantities are still cell averages. Whenever a cell-averaged magnetic field is required (for example, during the reconstruction stage), we use a linear interpolation:

$$(Bx)_n^\bullet = \frac{1}{2}\left[(bx)_n^{i*} + (bx)_n^{i*-1}\right], \tag{6.15}$$

where we now use the upper-case B to denote a cell-averaged magnetic field. Tóth (2000) proposed a simple extension to this idea: to build the interpolation into the algorithm itself, thus eliminating the need for a staggered grid. His paper only includes the equations for a 2D system, but as he mentions, it is straightforward to extend these to three dimensions

$$\begin{aligned}(Bx)_{n+1}^\bullet &= \frac{1}{2}\left[(bx)_{n+1}^{i*} + (bx)_{n+1}^{i*-1}\right] \\ &= (Bx)_n^\bullet - \frac{\Delta t}{\Delta y}\left[(Ez)_{\to y}^{j*} - (Ez)_{\to y}^{j*-1}\right] + \frac{\Delta t}{\Delta z}\left[(Ey)_{\to z}^{k*} - (Ey)_{\to z}^{k*-1}\right],\end{aligned} \tag{6.16}$$

where we use the definition of $(bx)_{n+1}^{i*}$ given in eq. (6.6). Similarly:

$$(By)_{n+1}^\bullet = (By)_n^\bullet - \frac{\Delta t}{\Delta z}\left[(Ex)_{\to z}^{k*} - (Ex)_{\to z}^{k*-1}\right] + \frac{\Delta t}{\Delta x}\left[(Ez)_{\to x}^{i*} - (Ez)_{\to x}^{i*-1}\right], \tag{6.17}$$

$$(Bz)_{n+1}^\bullet = (Bz)_n^\bullet - \frac{\Delta t}{\Delta x}\left[(Ey)_{\to x}^{i*} - (Ey)_{\to x}^{i*-1}\right] + \frac{\Delta t}{\Delta y}\left[(Ex)_{\to y}^{j*} - (Ex)_{\to y}^{j*-1}\right]. \tag{6.18}$$

A full derivation of this is given in Appendix B. The electric field terms are given by, e.g.:

$$(Ez)_{\to y}^{j*} = \frac{1}{2}\left[(Ez)_{n*}^{i*,j*} + (Ez)_{n*}^{i*-1,j*}\right], \tag{6.19}$$

which is a linear interpolation of two edge averages of E_z on the face $(j*)$. Note that $(Ez)_{\to y}^{j*}$ is distinct from $(Ez)_{\to x}^{i*}$, which is a linear interpolation on the face $(i*)$:

$$(Ez)_{\to x}^{i*} = \frac{1}{2}\left[(Ez)_{n*}^{i*,j*} + (Ez)_{n*}^{i*,j*-1}\right]. \tag{6.20}$$

The difference between these is illustrated in Fig. 6.1b. We write eqs. (6.16), (6.17) and (6.18) in this way to show that they are analogous to eqs. (6.6), (6.8) and (6.9). Substituting in eqs.

(6.12)-(6.14), we can write these electric field terms in terms of Riemann solver fluxes:

$$(Ex)^{j*}_{\rightarrow y} = \frac{1}{8}\left[(\tilde{H}7)^{j,k*}_{n*} + (\tilde{H}7)^{j+1,k*}_{n*} - (\tilde{G}8)^{j*,k}_{n*} - (\tilde{G}8)^{j*,k+1}_{n*}\right] + \frac{1}{8}\left[(\tilde{H}7)^{j,k*-1}_{n*} + (\tilde{H}7)^{j+1,k*-1}_{n*} - (\tilde{G}8)^{j*,k-1}_{n*} - (\tilde{G}8)^{j*,k}_{n*}\right], \tag{6.21}$$

$$(Ex)^{k*}_{\rightarrow z} = \frac{1}{8}\left[(\tilde{H}7)^{j,k*}_{n*} + (\tilde{H}7)^{j+1,k*}_{n*} - (\tilde{G}8)^{j*,k}_{n*} - (\tilde{G}8)^{j*,k+1}_{n*}\right] + \frac{1}{8}\left[(\tilde{H}7)^{j-1,k*}_{n*} + (\tilde{H}7)^{j,k*}_{n*} - (\tilde{G}8)^{j*-1,k}_{n*} - (\tilde{G}8)^{j*-1,k+1}_{n*}\right], \tag{6.22}$$

$$(Ey)^{i*}_{\rightarrow x} = \frac{1}{8}\left[(\tilde{F}8)^{i*,k}_{n*} + (\tilde{F}8)^{i*,k+1}_{n*} - (\tilde{H}6)^{i,k*}_{n*} - (\tilde{H}6)^{i+1,k*}_{n*}\right] + \frac{1}{8}\left[(\tilde{F}8)^{i*,k-1}_{n*} + (\tilde{F}8)^{i*,k}_{n*} - (\tilde{H}6)^{i,k*-1}_{n*} - (\tilde{H}6)^{i+1,k*-1}_{n*}\right], \tag{6.23}$$

$$(Ey)^{k*}_{\rightarrow z} = \frac{1}{8}\left[(\tilde{F}8)^{i*,k}_{n*} + (\tilde{F}8)^{i*,k+1}_{n*} - (\tilde{H}6)^{i,k*}_{n*} - (\tilde{H}6)^{i+1,k*}_{n*}\right] + \frac{1}{8}\left[(\tilde{F}8)^{i*-1,k}_{n*} + (\tilde{F}8)^{i*-1,k+1}_{n*} - (\tilde{H}6)^{i-1,k*}_{n*} - (\tilde{H}6)^{i,k*}_{n*}\right], \tag{6.24}$$

$$(Ez)^{i*}_{\rightarrow x} = \frac{1}{8}\left[(\tilde{G}6)^{i,j*}_{n*} + (\tilde{G}6)^{i+1,j*}_{n*} - (\tilde{F}7)^{i*,j}_{n*} - (\tilde{F}7)^{i*,j+1}_{n*}\right] + \frac{1}{8}\left[(\tilde{G}6)^{i,j*-1}_{n*} + (\tilde{G}6)^{i+1,j*-1}_{n*} - (\tilde{F}7)^{i*,j-1}_{n*} - (\tilde{F}7)^{i*,j}_{n*}\right], \tag{6.25}$$

$$(Ez)^{j*}_{\rightarrow y} = \frac{1}{8}\left[(\tilde{G}6)^{i,j*}_{n*} + (\tilde{G}6)^{i+1,j*}_{n*} - (\tilde{F}7)^{i*,j}_{n*} - (\tilde{F}7)^{i*,j+1}_{n*}\right] + \frac{1}{8}\left[(\tilde{G}6)^{i-1,j*}_{n*} + (\tilde{G}6)^{i,j*}_{n*} - (\tilde{F}7)^{i*-1,j}_{n*} - (\tilde{F}7)^{i*-1,j+1}_{n*}\right]. \tag{6.26}$$

This concludes the description of the Balsara-Spicer-Tóth algorithm.

6.2.4 Disadvantages of this Algorithm

The simplest implementation of this algorithm involves creating six new 3D arrays, one for each of the intercell fluxes we require: $(\tilde{F}7)$, $(\tilde{F}8)$, $(\tilde{G}6)$, $(\tilde{G}8)$, $(\tilde{H}6)$ and $(\tilde{H}7)$. Every time we complete the solve stage for a given 1D row, we obtain the values of two of these fluxes along the entire row, and so can store them in the appropriate arrays. Once a complete set of three directional passes has been completed, we can sort these arrays into another six 3D arrays, one for each of the electric field terms: $(Ex)^{j*}_{\rightarrow y}$, $(Ex)^{k*}_{\rightarrow z}$, $(Ey)^{i*}_{\rightarrow x}$, $(Ey)^{k*}_{\rightarrow z}$, $(Ez)^{i*}_{\rightarrow x}$ and $(Ez)^{j*}_{\rightarrow y}$. This is done according to eqs. (6.21)-(6.26) above. Finally, we use these components to update the magnetic field quantities using eqs. (6.16)-(6.18). This algorithm has been shown to be reliable through standard MHD tests (Tóth, 2000), but has some notable disadvantages:

1. *Additional storage of information.* In a DS-RSA code: we extract a single row from the 3D grid, evolve all of the state quantities along that row, and then return it to the grid. We do not need to store any information from that row to evolve other rows in the same directional pass. This is not the case here: we need to store all the intercell fluxes.

2. *Additional processing between sets of directional passes.* In a DS-RSA code: once a complete set of three directional passes has been completed, the state quantities require no further processing until the next set. Again, this is not the case here: we need to carry out additional operations on the intercell fluxes and electric field components in order to determine the new magnetic field quantities.

3. *Staggered grids.* Although we use a cell-averaged magnetic field, we still require staggered grids: the intercell fluxes and electric field terms are stored as *face* averages. This complicates their implementation, especially with adaptive grids.

4. *Limited parallelisation.* Because of the extra processing described above, and the fact that the electric field terms depend on intercell fluxes from different rows *and* directional passes, it is not immediately obvious how this algorithm could be parallelised.

While these are not major issues, they complicate the implementation of this algorithm in existing, mature astrophysical codes. We would therefore prefer an algorithm that: minimises the data that needs to be stored from each row during the directional passes; reduces the amount of processing required between complete sets of directional passes; completely removes the need for any staggered grids; simplifies and maximises the possible parallelisation. We now show that such an algorithm exists, by reformulating the Balsara-Spicer-Tóth algorithm so that it is more compatible with the DS-RSA approach.

6.3 A Modified Approach

To start, we define just three new 3D arrays: $\Delta(Bx)$, $\Delta(By)$ and $\Delta(Bz)$. After each complete set of three directional passes, we aim for these to contain the change in the magnetic field during those passes. To make our derivation clearer, we focus on a single cell: (i, j, k).

6.3.1 The x-Direction Pass

Contribution to $\Delta(Bx)^\bullet$

The change in $\Delta(Bx)^\bullet$ during each timestep is given by eq. (6.16), and depends on four electric field components: $(\bar{E}z)^{j*}_{\rightarrow y}$, $(\bar{E}z)^{j*-1}_{\rightarrow y}$, $(\bar{E}y)^{k*}_{\rightarrow z}$ and $(\bar{E}y)^{k*-1}_{\rightarrow z}$. We can substitute in the definition of these components, given by eqs. (6.24) and (6.26). However, we only include the $\tilde{\mathbf{F}}$ terms, as these are the only ones we obtain during the x-direction pass. The full derivation

is given in Appendix B, but we are eventually left with the following expression:

$$\Delta(Bx)^{\bullet} := \Delta(Bx)^{\bullet} + \frac{\Delta t}{\Delta y}\frac{1}{8}\left[(\tilde{F}7)^{i*,j+1}_{n*} + (\tilde{F}7)^{i*-1,j+1}_{n*} - (\tilde{F}7)^{i*,j-1}_{n*} - (\tilde{F}7)^{i*-1,j-1}_{n*}\right] + \frac{\Delta t}{\Delta z}\frac{1}{8}\left[(\tilde{F}8)^{i*,k+1}_{n*} + (\tilde{F}8)^{i*-1,k+1}_{n*} - (\tilde{F}8)^{i*,k-1}_{n*} - (\tilde{F}8)^{i*-1,k-1}_{n*}\right], \quad (6.27)$$

where use ':=' to refer to an update in the usual computational sense.

This is still not in the spirit of dimensionally-split algorithms, as the right-hand side of the expression depends on fluxes from several different rows, as shown by the different spatial indices. Instead, we want to consider what we could do after a *given* x-row – labelled (i, J, K) with fixed J, K – has been evolved. To do this, we use the fact that if $\Delta(Bx)^{\bullet}$ depends on, say, $(\tilde{F}7)^{i*,j+1}_{n*}$, then this is equivalent to saying that $\Delta(Bx)^{j-1}$ depends on $(\tilde{F}7)^{i*}_{n*}$. If we consider each term in eq. (6.27) in this manner, then we can rewrite it as:

$$\Delta(Bx)^{J\pm1} := \Delta(Bx)^{J\pm1} \mp \frac{\Delta t}{\Delta y}\frac{1}{8}\left[(\tilde{F}7)^{i*}_{n*} + (\tilde{F}7)^{i*-1}_{n*}\right], \quad (6.28)$$

$$\Delta(Bx)^{K\pm1} := \Delta(Bx)^{K\pm1} \mp \frac{\Delta t}{\Delta z}\frac{1}{8}\left[(\tilde{F}8)^{i*}_{n*} + (\tilde{F}8)^{i*-1}_{n*}\right]. \quad (6.29)$$

Both the right-hand sides only contain terms from the row we are currently on, $(i*)$. This means that once a given row has been evolved, we can process the contribution of the inter-cell fluxes to the magnetic field quantities immediately – using eqs. (6.28) and (6.29) – without needing to store them in any other form. It is also worth noting that these equations are of a similar form to the standard RSA update given by eq. (2.17).

Contribution to $\Delta(By)^{\bullet}$ and $\Delta(Bz)^{\bullet}$

Similarly, we use eq. (6.17) to find the contribution to $\Delta(By)^{\bullet}$ during the x-direction pass:

$$\Delta(By)^{\bullet} := \Delta(By)^{\bullet} + \frac{\Delta t}{\Delta x}\frac{1}{4}\left[-(\tilde{F}7)^{i*}_{n*} + (\tilde{F}7)^{i*-1}_{n*}\right], \quad (6.30)$$

$$\Delta(By)^{j\pm1} := \Delta(By)^{j\pm1} + \frac{\Delta t}{\Delta x}\frac{1}{8}\left[-(\tilde{F}7)^{i*}_{n*} + (\tilde{F}7)^{i*-1}_{n*}\right], \quad (6.31)$$

while eq. (6.18) gives us the contribution to $\Delta(Bz)^{\bullet}$ during the x-direction pass:

$$\Delta(Bz)^{\bullet} := \Delta(Bz)^{\bullet} + \frac{\Delta t}{\Delta x}\frac{1}{4}\left[-(\tilde{F}8)^{i*}_{n*} + (\tilde{F}8)^{i*-1}_{n*}\right], \quad (6.32)$$

$$\Delta(Bz)^{k\pm1} := \Delta(Bz)^{k\pm1} + \frac{\Delta t}{\Delta x}\frac{1}{8}\left[-(\tilde{F}8)^{i*}_{n*} + (\tilde{F}8)^{i*-1}_{n*}\right]. \quad (6.33)$$

6.3.2 The y- and z-Direction Passes

We go through the same steps as above for the y-direction pass, but start by considering $\Delta(By)^{\bullet}$. The change in this during each timestep is given by eq. (6.17), and depends on the electric-field components $(\bar{E}x)^{k*}_{\to z}$, $(\bar{E}x)^{k*-1}_{\to z}$, $(\bar{E}z)^{i*}_{\to x}$ and $(\bar{E}z)^{i*-1}_{\to x}$. Again, we substitute in the definition of these components, given by eqs. (6.22) and (6.25), although this time we only

include the $\tilde{\mathbf{G}}$ terms. After simplification, we are left with:

$$\Delta(By)^{\bullet} := \Delta(By)^{\bullet} + \frac{\Delta t}{\Delta x}\left[(\tilde{G}6)_{n*}^{i+1,j*} + (\tilde{G}6)_{n*}^{i+1,j*-1} - (\tilde{G}6)_{n*}^{i-1,j*} - (\tilde{G}6)_{n*}^{i-1,j*-1}\right] + \frac{\Delta t}{\Delta z}\left[(\tilde{G}8)_{n*}^{j*,k+1} + (\tilde{G}8)_{n*}^{j*-1,k+1} - (\tilde{G}8)_{n*}^{j*,k-1} - (\tilde{G}8)_{n*}^{j*-1,k-1}\right]. \quad (6.34)$$

Before we continue, a note on dimensionally-split codes: recall that the way we extract the components of the linear momentum depends on what the directional pass is. (This is summarised by Table 2.1 in Chapter 2). This is so that we do not have to rewrite the 1D code for each directional pass. The same approach applies to the components of the magnetic field. If we rewrite the terms in eq. (6.34) according to this approach – such that $\{x, F, i, 6\} \rightarrow \{y, G, j, 7\}$ and $\{y, G, j, 7\} \rightarrow \{x, F, i, 6\}$ – then we actually retrieve eq. (6.27). Furthermore, it can be shown that *all* of eqs. (6.28)-(6.33) can be retrieved in this manner, and for the z-direction pass as well. In other words, once the equations for the x-direction pass has been implemented, we do not need to re-write them for the y- and z-direction passes, as long as the Cartesian components are extracted and returned according to the DS approach.

6.3.3 Final Update

Once all three passes have been completed, we simply perform the following operation:

$$(Bx) := (Bx) + \Delta(Bx) \quad (6.35)$$

and similarly for (By) and (Bz) to update the magnetic field. There are some additional subleties associated with this, regarding the pressure and energy (which both have a dependence on the magnetic field), and these are described in Section 6.4 below.

6.3.4 Comparison with the Balsara-Spicer-Tóth Algorithm

1. *Reduced storage of information.* We have minimised the amount of information that needs to be stored during the directional passes. Instead of storing the individual intercell fluxes and electric field components (which requires twelve 3D arrays), we only store the change in the magnetic field quantities (which requires just three).

2. *Reduced processing between sets of directional passes.* We have also minimised the amount of additional processing required between complete sets of directional passes: we only need to perform the simple update given in eq. (6.35).

3. *No staggered grids.* The $\Delta(Bx)$, $\Delta(By)$ and $\Delta(Bz)$ arrays store cell-averaged values, which means we no longer need to store any face-averaged values. This eliminates the need for staggered grids, which means this algorithm is easier to use with adaptive grids.

4. *Simplified parallelisation.* This algorithm is more easily parallelisable. For example, assume that we have a number of processors, each independently evolving some rows in a given directional pass. Each of these processors should have a copy of the $\Delta\mathbf{B}$ arrays,

which are updated as described above. Because these all contain *changes* in the magnetic field quantities, we only require a simple reduction operation to sum over all of these copies and obtain the total change before completing eq. (6.35).

5. *Dimensionally-split framework.* Finally, the row-by-row approach means that the algorithm is considerably simpler to implement in existing DS-RSA codes.

6.4 The Complete Algorithm

We will now describe the complete algorithm, with MAGNETO codes in square brackets. We use the following notation for the standard arrays required for the DS-RSA approach:

1. $7 \times \mathbf{U}^{3D}$: the 3D arrays which store the cell-averaged state quantities.
2. $7 \times \mathbf{U}^{1D}$: the 1D arrays which are used to store 1D rows extracted from the full 3D grid.

Additionally, we require two more sets of arrays for the magnetic field quantities:

3. $3 \times \mathbf{B}^{copy}$: stores a copy of the cell-averaged magnetic field at the start of each timestep.
4. $3 \times \Delta\mathbf{B}$: stores the change in the cell-averaged magnetic field during each timestep.

6.4.1 Before the Directional Passes

a. *Make a copy of the magnetic field quantities.* Copy the three $\mathbf{U}^{3D}$ arrays that store the cell-averaged magnetic field quantities into the corresponding $\mathbf{B}^{copy}$ arrays. These copies should not be modified during the directional passes. `[GRI03]`

6.4.2 The x-Direction Pass

b. *Extract a 1D row.* Choose an x-direction row from the grid, which is defined as having fixed j and k indices. Extract the cell-averaged state quantities along this row from the $\mathbf{U}^{3D}$ arrays, and store them in the corresponding $\mathbf{U}^{1D}$ arrays. `[GRI04]`

c. *Reconstruct: find the left and right states.* For each $\mathbf{U}^{1D}$ array, use a reconstruction algorithm to obtain the left and right states at each cell interface $(i*)$. `[REC02-REC05]`

d. *Solve: find the intercell fluxes.* Part (c) defines a series of Riemann problems. Use a Riemann solver algorithm to obtain the intercell fluxes, $\tilde{\mathbf{F}}$, at each cell interface. `[SOL02]`

e. *Update the state variables.* Use these intercell fluxes to update the cell averages stored in $\mathbf{U}^{1D}$ using the standard RSA update, given by eq. (2.17). Note that this should be done for *all* state quantities, including the magnetic field components. `[EV006]`

f. *Return the 1D row.* Return the new cell averages stored in the $\mathbf{U}^{1D}$ arrays to the $\mathbf{U}^{3D}$ arrays, and update the $\Delta\mathbf{B}$ arrays using eqs. (6.28)-(6.33). `[GRI08]`

g. *Complete the x-direction pass:* repeat steps (b) to (f) for all 1D rows in the x-direction.

6.4.3 The y- and z-Direction Passes

h. *Complete all passes:* repeat the equivalents of steps (b) to (g) for the y- and z- directional passes. Remember to extract and return the linear momentum and magnetic field components in the usual dimensionally-split manner, as described in Table 2.1 in Chapter 2. Also note that in some code implementations – including ours – the order of the passes is alternated between timesteps to preserve the accuracy of the 1D method.

6.4.4 After the Dimensional Passes

After all of the above steps have been completed, the fluid quantities will be fully evolved to the next timestep. We will also have three versions of the magnetic field quantities:

- Their correct values at the start of the timestep (stored in $\mathbf{B}^{\mathrm{s}}$);
- Their correct change in value during the timestep, as determined by the constrained transport algorithm (stored in $\Delta\mathbf{B}$);
- Their incorrect values at the end of the timestep, as determined by the standard RSA strategy (stored in the $\mathbf{U}^{3\mathrm{D}}$ arrays corresponding to the magnetic field quantities).

Then the final steps are:

i. *Correct the energy and pressure (Part I).* This is an optional step. We use the magnetic field quantities stored in the $\mathbf{U}^{3\mathrm{D}}$ arrays to calculate the magnetic field pressure p_{M} (using $p_{\mathrm{M}} = \frac{1}{2}\mathbf{B}^2$). We then subtract this contribution from the newly evolved values of the total energy and the total pressure. `[EVO07]`

j. *Update the magnetic field.* We replace the magnetic field quantities stored in the $\mathbf{U}^{3\mathrm{D}}$ arrays with $\mathbf{B}^{\mathrm{s}} + \Delta\mathbf{B}$ (in other words, their value at the start of the timestep, plus their *correct* change in value during the timestep, according to the constrained transport algorithm). Consequently, the stored magnetic field quantities are now evolved for one timestep and maintain the divergence-free condition. `[EVO07]`

k. *Correct the energy and pressure (Part II).* If step (i) was completed: use these new magnetic field quantities to recalculate the magnetic pressure and add this back to the total energy and total pressure. This correction means that energy may not be conserved entirely (as discussed in Balsara and Spicer, 1999), but avoids negative gas pressures being calculated if $p_{\mathrm{M}} \gg p_{\mathrm{G}}$, as described in Section 6.4.5. `[EVO07]`

6.4.5 Some Caveats

The parallel component of the magnetic field. in a 1D system, the parallel component of the magnetic field – which is (Bx) in the x-direction pass – should be uniform and constant in time. This automatically satisfies the divergence-free condition. As we saw in the previous chapter, the reconstruction and solver algorithms we use were designed for such 1D systems, which means that the former does not find left and right states for this parallel component in step (c), while the latter assumes it is constant across the cell interface in step (d). This is not an

issue when using staggered grids, because this constant value is stored explicitly at the cell interface. However, this is clearly not the case for non-staggered grids, which means we must interpolate this parallel component to get its value across the interface. Our tests have shown that a simple linear interpolation is sufficient, for example: $(Bx)^{\text{int},i*} = [(Bx)^i + (Bx)^{i+1}]/2$.

The gas pressure. during each directional pass, we still update the magnetic field quantities using the (incorrect) RSA strategy. This is because we require the *gas* pressure p_{G} when we return to the earlier steps in other directional passes. However, recall that we only store the total pressure p, so we need to calculate it by subtracting the magnetic pressure p_{M} (i.e.$p_{\text{G}} = p - p_{\text{M}}$), which is in turn determined by the magnetic field quantities. Hence, in cases where $p_{\text{M}} \gg p_{\text{G}}$, not updating the magnetic field components can result in negative pressures. This is the same reason why we make the pressure corrections in steps (i) and (k). This issue is discussed further in Balsara and Spicer (1999).

6.5 2D MHD Tests

To illustrate the accuracy of the new algorithm, we conclude by presenting the results of four standard MHD tests. All results were obtained using the second-order TVD reconstruction of Balsara (1998), and the HLLD Riemann solver of Miyoshi and Kusano (2005), as described in Chapter 5. Initial conditions are given for the mass density ρ, the gas pressure p_G, the velocity $\mathbf{v} = (v_x, v_y, v_z)$, the adiabtic index γ and the magnetic field $\mathbf{B} = (B_x, B_y, B_z)$. Although these are 2D tests, the simulation is run in 3D with one cell in the z-direction. For all tests, the divergence was found to be zero (to machine precision) at the end of the simulation time.

6.5.1 Circularly-Polarised Alfvén Waves

The test of circularly polarised Alfvén waves were first described by Tóth (2000). Such waves are smooth, analytic, non-linear solutions of the MHD equations, which makes them ideal for testing the implementation of these equations. The version we test is sinusoidal and uses a periodic domain that can fit exactly one wavelength. We test two versions: a travelling wave, where the wave returns to its original position at integer multiples of the period (which is equal to 1 in this case), and a standing wave, where it stays at its original position.

This test is implemented as subroutine `[TES05]`in Magneto (see previous chapter for details of this code). We use the initial conditions described in Tóth (2000). We start by defining a rotated coordinate system, $c_{\parallel}$ and $c_{\perp}$. These are rotated by an angle $\theta = \pi/6$ from the Cartesian x and y (and hence our discretised grid) such that:

$$c_{\parallel} = x\cos\theta + y\sin\theta \quad \text{and} \quad c_{\perp} = -x\sin\theta + y\cos\theta \tag{6.36}$$

We define the velocity and magnetic field components along these axes: $v_{\parallel} = 0$ for travelling waves and $v_{\parallel} = 1$ for standing waves; $B_{\parallel} = 1$; $v_{\perp} = B_{\perp} = 0.1\sin(2\pi c_{\parallel})$. We then have:

`Circularly-Polarised Alfvén Waves`

- **The Grid**
  - Domain: $0 < x < 1/\cos\theta$, $0 < y < \sin\theta$;
  - Number of cells: varying from 8^2 to 32^2;
  - Total time: $t = 5$.
- **Fluid Quantities**
  - $\rho = 1$;
  - $p_G = 0.1$;
  - $v_x = v_\parallel \cos\theta - v_\perp \sin\theta$;
  - $v_y = v_\parallel \sin\theta + v_\perp \cos\theta$;
  - $v_z = 0.1\cos(2\pi c_\parallel)$;
  - $\gamma = 5/3$.
- **Magnetic Field Quantities**
  - $B_x = B_\parallel \cos\theta - B_\perp \sin\theta$;
  - $B_y = B_\parallel \sin\theta + B_\perp \cos\theta$;
  - $B_z = 0.1\cos(2\pi c_\parallel)$.

The results for both the travelling and standing waves at time $t = 5$ are shown in Fig. 6.2, and are comparable with figs. 8 and 9 in Tóth (2000). Note that we plot $B_\perp$, which can be determined from B_x and B_y using eq. (6.36). We include reference plots, which are simply the initial conditions on a grid with 128×128 cells. This was designed as a difficult test – especially at low resolutions – so the results do not coincide exactly with the expected analytic solution. However, they are comparable with results from other algorithms, as presented in Tóth (2000). It may also be possible to improve them with a higher-order reconstruction.[2]

6.5.2 The Blast Problem

The blast problem is initialised as a small disk of high-pressure fluid in the centre of the grid, which then expands rapidly into the low-pressure ambient fluid. As pointed out by Balsara and Spicer (1999), it is not necessarily useful for testing whether the magnetic field is divergence free: the conditions are such that the build-up of non-zero $\nabla \cdot \mathbf{B}$ will not have a major effect on the overall dynamics. However, it is useful for testing the propagation of strong MHD shocks in multiple dimensions (Stone *et al.*, 2008).

This test is implemented as subroutine `[TES06]` in MAGNETO. We use the initial conditions described in Stone *et al.* (2008) (which were in turn taken from Londrillo and del Zanna, 2000):

[2] The published results presented here were completed before the results presented in Chapter 5, and so we use the TVD reconstruction of Balsara (1998), rather than the higher-order PPM reconstruction of Stone *et al.* (2008).

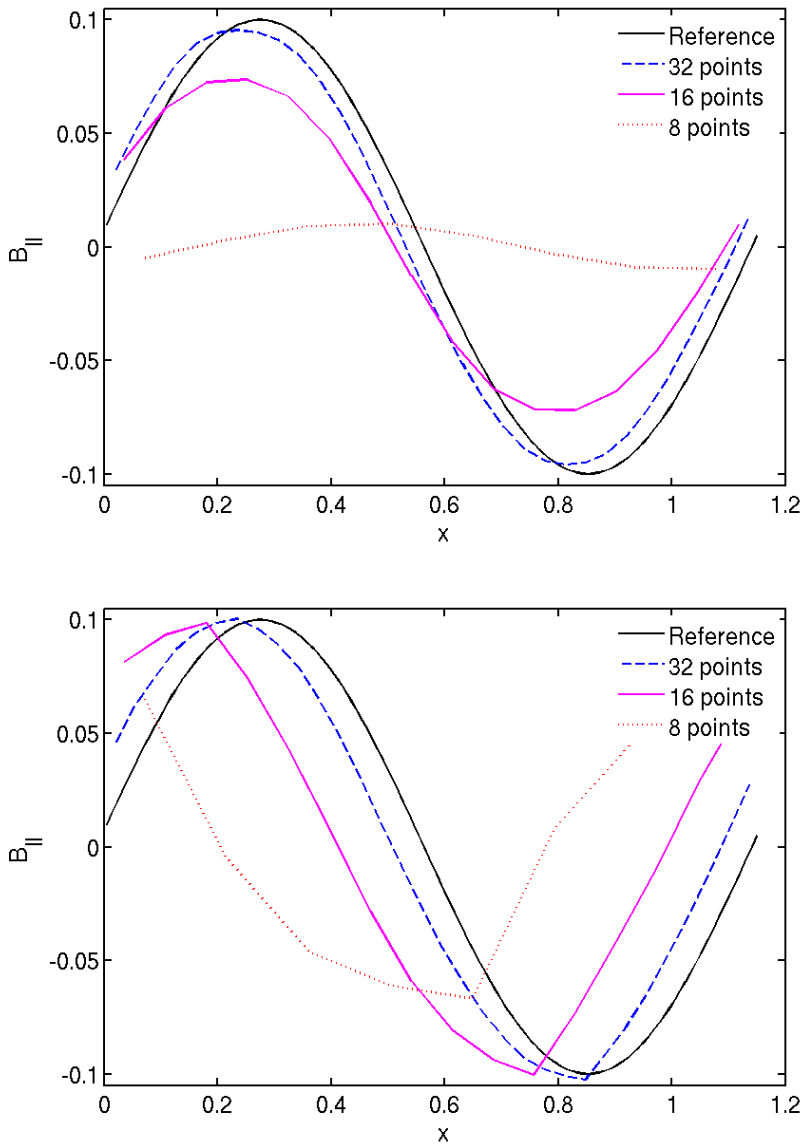


Figure 6.2: Plot of $B_\perp = (\sqrt{3}B_y - B_x)/2$, from the circularly-polarised Alfvén waves test at time $t = 5$, initialised as described in section 6.5.1. Top: results for travelling waves, with parallel velocity $v_\parallel = 0$. Bottom: results for standing waves, with parallel velocity $v_\parallel = 1$.

The Blast Problem

- **The grid**
  - Domain: $-0.5 < x < 0.5$; $-0.75 < y < 0.75$;
  - Number of cells: 300×450;
  - Total time $t = 0.2$.
- **Fluid quantities**
  - $\rho = 1$;
  - $p_G = 10$ within a radius $r < 0.1$ (equivalent to 30 cells) in the centre of the grid, and $p_G = 0.1$ everywhere else.
  - $\mathbf{v} = 0$;
  - $\gamma = 5/3$;
- **Magnetic field quantities**
  - $B_x = B_y = 1/\sqrt{2}$;
  - $B_z = 0$.

The mass density, thermal pressure and magnetic pressure at time $t = 0.2$ are shown in Fig. 6.3, and are comparable with fig. 28 in Stone *et al.* (2008). The results are as expected: the blast – which would otherwise be circular - has aligned itself with the magnetic field. The near-perfect π-rotational symmetry of the results are also of note.

6.5.3 The Orszag-Tang Vortex

The vortex of Orszag and Tang (1979) is another commonly-used 2D MHD test, which starts with smooth initial data but becomes progressively more complex (Tóth, 2000). The expected symmetry of the solution is useful for showing that the dimensionally-split algorithm (which uses the same code for each dimensional pass, but just extracts and returns the rows differently) works and has been implemented correctly.

This test is implemented as subroutine `[TES07]` in MAGNETO. We use the initial conditions described in Stone *et al.* (2008) (which were in turn taken from Orszag and Tang, 1979).

The Orszag-Tang Vortex

- **The grid**
  - Domain: $0 < x, y < 1$;
  - Number of cells: 128^2;
  - Total time $t = 0.5$.
- **Fluid quantities**
  - $\rho = 25/36\pi$;
  - $p_G = 5/12\pi$;
  - $v_x = -\sin(2\pi y)$;

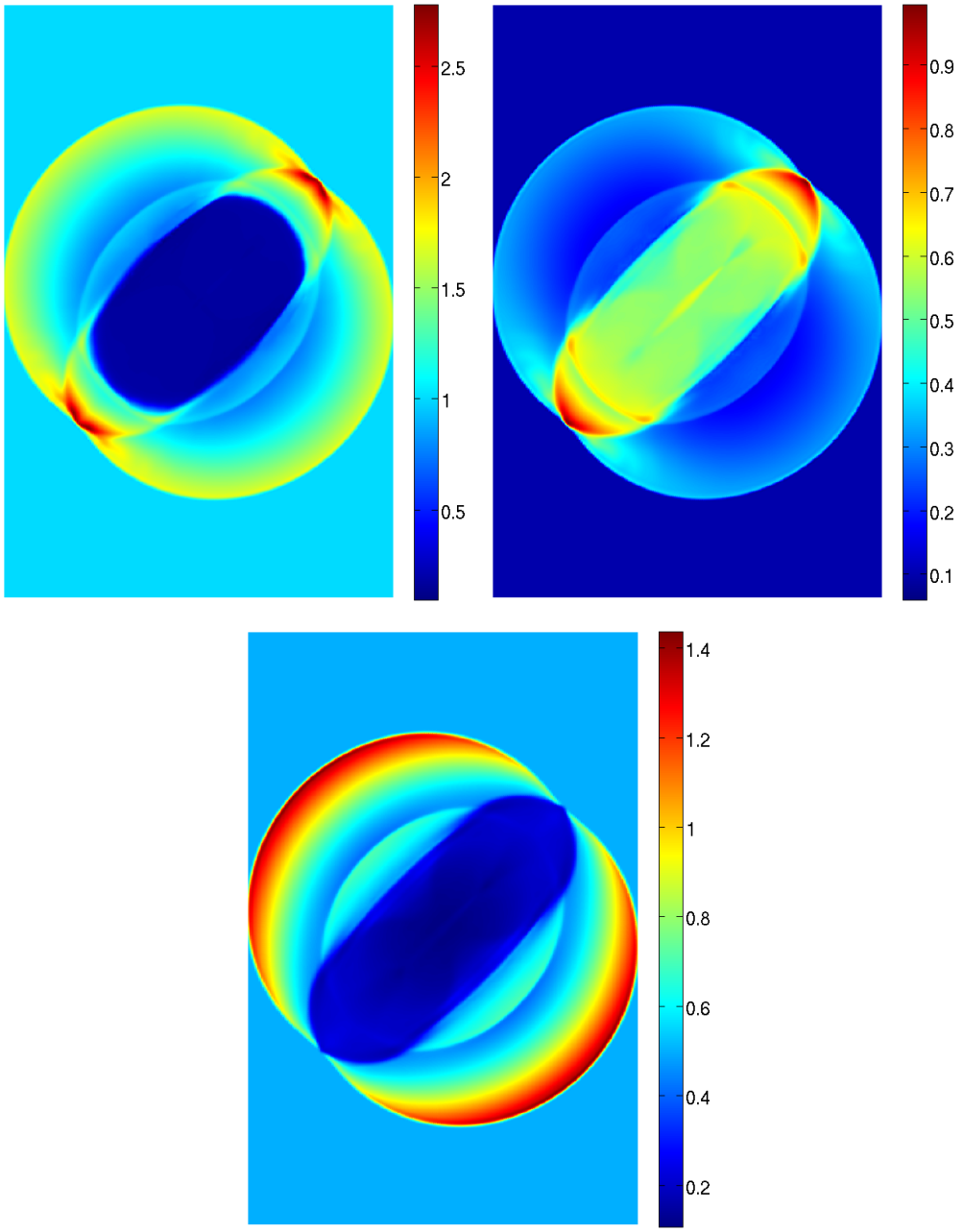


Figure 6.3: Results from the blast problem at $t = 0.2$, initialised as described in section 6.5.2. Top left: mass density. Top right: thermal pressure. Bottom: magnetic pressure.

- $v_y = \sin(2\pi x)$;
- $v_z = 0$;
- $\gamma = 5/3$.

- **Magnetic field quantities**
  - $B_x = -\sin(2\pi y)/\sqrt{4\pi}$;
  - $B_y = \sin(4\pi x)/\sqrt{4\pi}$;
  - $B_z = 0$.

The mass density, thermal pressure and magnetic pressure at time $t = 0.5$ are shown in Fig. 6.4, and are comparable with fig. 22 in Stone *et al.* (2008). The mass density can also be compared with fig. 16 in Tóth (2000) or fig. 14 in Dai and Woodward (1998). The π-rotational symmetry of the results implies that our dimensionally-split approach is effective.

6.5.4 The Rotor Problem

Finally, we consider the rotor problem, first proposed in its current form by Balsara and Spicer (1999). This involves placing a rotating disk in the centre of a square domain, which creates rotational discontinuities and propagates Alfvén waves into the ambient fluid. The magnetic field is initialised such that it wraps around the rotor, squeezing the fluid from its original circular form. This problem is very useful for testing whether the magnetic field is divergence free, because the build-up of non-zero $\nabla \cdot \mathbf{B} = 0$ greatly influences the dynamics.

This test is implemented as subroutine `[TES08]` in MAGNETO. We use the initial conditions described in Balsara and Spicer (1999). However, as noted by Tóth (2000), the results they present are (erroneously) from a different set of conditions. We stick to what is described in the text, and compare our results with the correct results presented in other papers. We define the radius of the rotor in the centre of the grid as $r < r_0 = 0.1$. This is separated from the ambient fluid by a taper, $r_0 < r < r_1 = 0.115$. We also define $v_0 = 2$, and a constant $f = (r_1 - r)/(r_1 - r_0)$ (which is used for the taper). Then:

`The Rotor Problem`

- **The grid**
  - Domain: $-0.5 < x, y < 0.5$;
  - Number of cells: 256^2;
  - Total time $t = 0.15$.

- **Fluid quantities: ambient fluid**
  - $\rho = 1$;
  - $p_G = 1$;
  - $\mathbf{v} = 0$;
  - $\gamma = 1.4$.

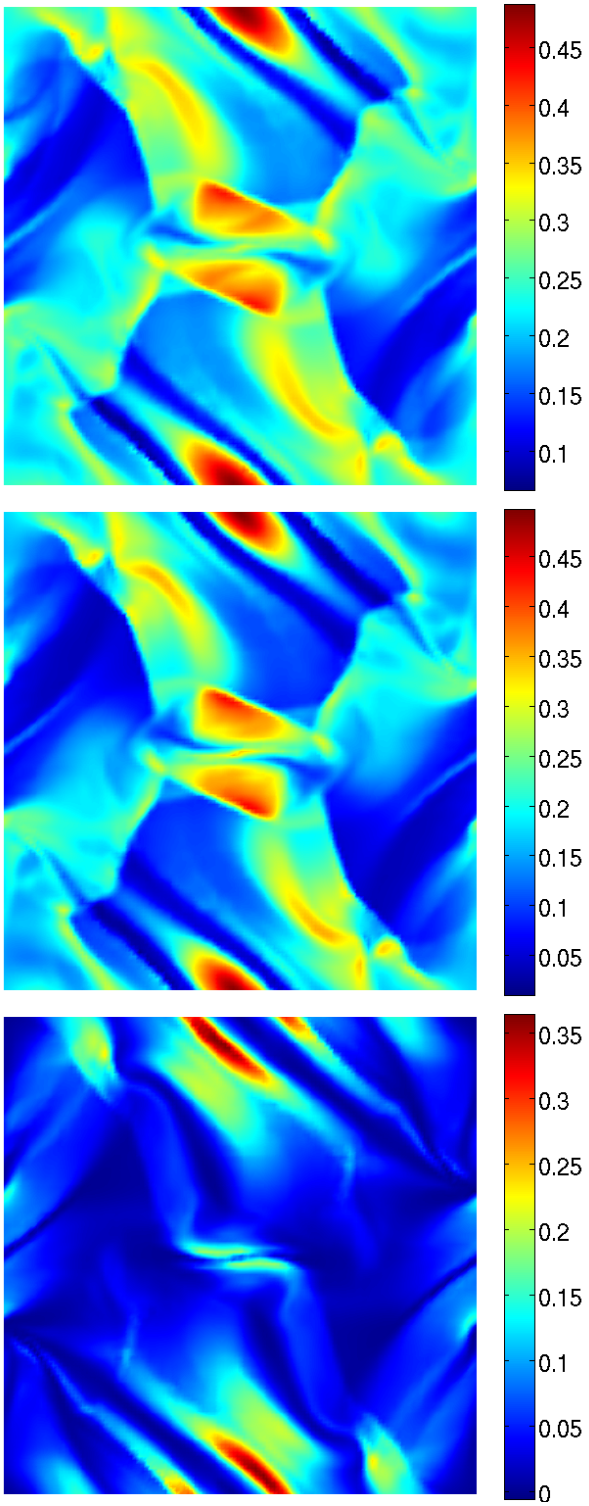


Figure 6.4: Results from the Orszag-Tang vortex at $t = 0.5$, initialised as in section 6.5.3. Top mass density. Middle: thermal pressure. Bottom: magnetic pressure.

- **Fluid quantities: the rotor**

  which is defined within a radius $r < r_0 = 0.1$ (~ 26 cells)
  - $\rho = 10$;
  - $v_x = -v_0(y - 0.5)/r_0$;
  - $v_y = v_0(x - 0.5)/r_0$.
- **Fluid quantities: the taper**

  which is defined within the band $r_0 < r < r_1 = 0.115$
  - $\rho = 1 + 9f$;
  - $v_x = -fv_0(y - 0.5)/r$;
  - $v_y = fv_0(x - 0.5)/r$.
- **Magnetic field quantities**
  - $B_x = 5$;
  - $B_y = B_z = 0$.

The mass density, thermal pressure and magnetic pressure at time $t = 0.15$ are shown in Fig. 6.5. In each plot, there are thirty equally-spaced contours between the highest and lowest values. These results are comparable with fig. 18 of Tóth (2000) or fig. 25 of Stone *et al.* (2008), and imply that the magnetic field is divergence free. Again, it is worth noting the π-rotational symmetry of the results.

6.6 Conclusions

In this chapter, we presented a new, partially dimensionally-split algorithm for numerically solving the MHD equations. Specifically, we derived a variation of the algorithm of Balsara and Spicer (1999) and Tóth (2000). It is based on the standard reconstruct-solve-average strategy, and relies on constrained transport to ensure that the magnetic field remains divergence free ($\nabla \cdot \mathbf{B} = 0$). Our approach restores many of the advantages of a dimensionally-split reconstruct-solve-average scheme, which were lost by the Balsara-Spicer-Tóth algorithm:

- The information stored during the directional passes is minimised.
- The processing between complete sets of three directional passes is minimised.
- The need for staggered grids has been completely eliminated.
- The algorithm is more easily parallelisable.

Furthermore, all of these advanages considerably simplify the development required to add this algorithm – and hence magnetic fields – to Charybdis. Finally, we demonstrated the accuracy and robustness of the algorithm with four standard MHD test problems.

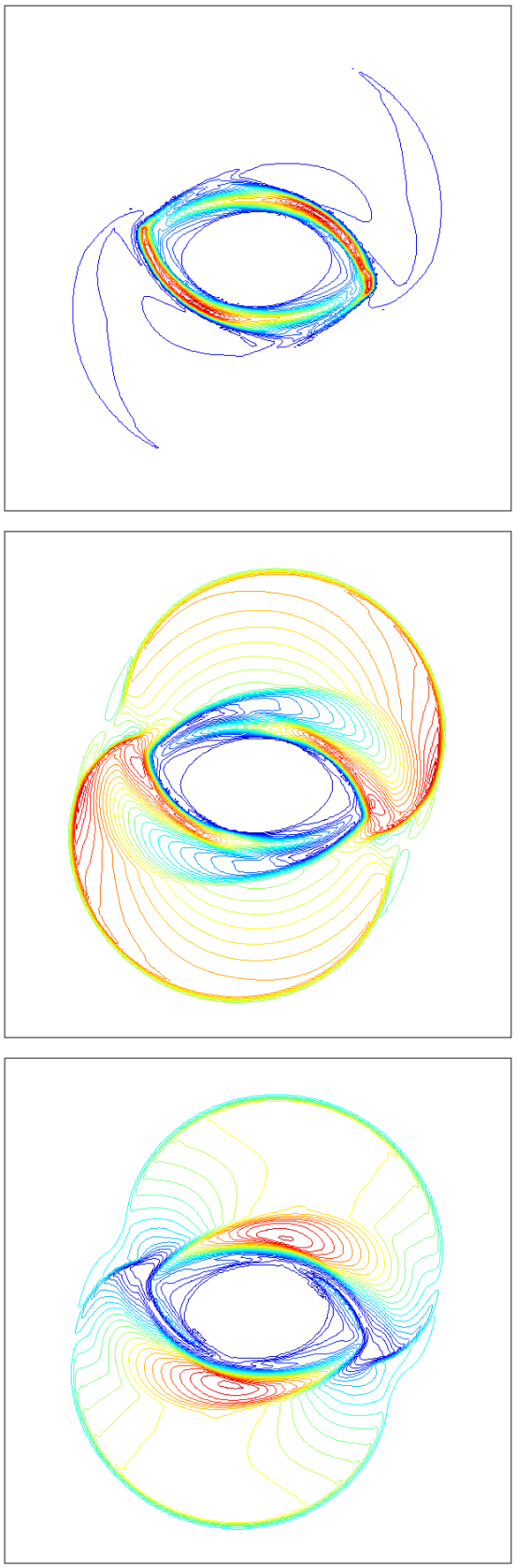


Figure 6.5: Results from the rotor problem at $t = 0.15$, initialised as in section 6.5.4. All contours are linearly-spaced from 0.426 (blue) to 12.4 (red) for the mass density (top), 0.0369 (blue) to 1.98 (red) for the thermal pressure (middle) and 0.0154 (blue) to 2.60 (red) for the magnetic pressure (bottom).

Chapter 7
Towards Magnetised BH-NS Mergers

In this chapter, we begin upgrading Charybdis to include magnetic field effects, by implementing the algorithms described in the previous two chapters. Note that this work was not completed in time for inclusion in this thesis. We report on the progress so far, and comment on open problems for future investigations.

7.1 Adding Magnetic Field Quantities

The first step was to go through all the subroutines and functions of Charybdis, and add the three components of the magnetic field where necessary (B_x, B_y and B_z). As well as the hydrodynamics engine, this included e.g. initialisation and data output. Helpfully, this could be done using the three components of the velocity as reference. Then:

- The PPM algorithm of Colella and Woodward (1984) was replaced with the newer (characteristic) version of Stone *et al.* (2008), which was described in Chapter 5. This required new subroutines for calculating the MHD eigensystem.
- The exact Riemann solver was replaced with the HLLD approximate Riemann solver, which was also described in Chapter 5. This only models the conserved variables of the MHD equations, so the advection equations described in Section 3.1.2 were also added.
- The new constrained transport algorithm was implemented, which was described in Chapter 6. This was done using the steps given in Section 6.4.
- Any energy and pressure calculations were modified to take include magnetic contributions. For example: Charybdis only stores the total energy, but the EOS sometimes requires the internal energy. Previously, this was calculated by subtracting the kinetic energy from the total energy, but now the magnetic energy must also be subtracted.
- The calculation of new timesteps – based on the Courant-Friedrich-Lewy condition described in Section 2.5 – was modified to include magnetic wavespeeds.

Short tests were run after each stage, to minimise bugs. Additionally, the HLLD and PPM algorithms were copied from the test code Magneto, where they have been shown to work. We then ran a longer test simulation, with all the magnetic field components set to zero. This

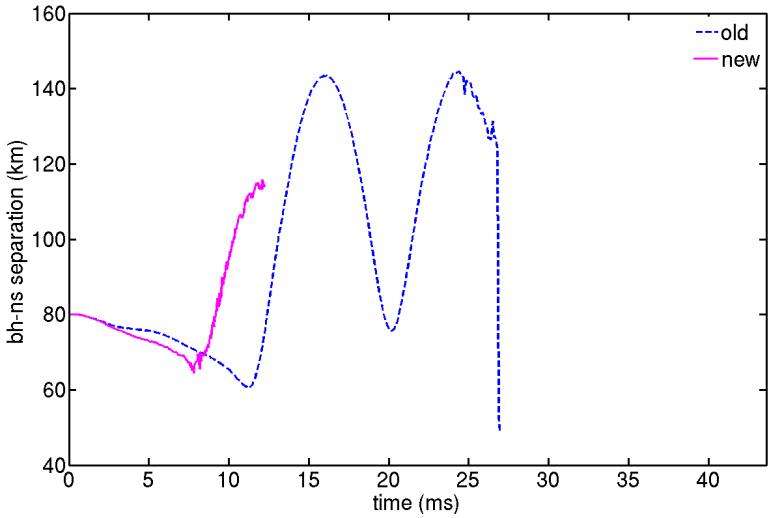


Figure 7.1: BH-NS separation as a function of time, for the old code (blue/dotted) and the new version (magenta/solid) with all the implemented changes as described in the main text.

is a necessary (but not sufficient) test to check whether the above changes were implemented correctly.[1] We repeated the G11-NL3-3 run used in Chapter 4, where the mass ratio $q = 3$, the black hole spin $s = 0.5$, and the initial separation $d = 80$ km.

Unfortunately, one of the changes – which has not yet been identified – caused the timesteps to be significantly reduced. This was true even when the changes made to the timestep subroutine were reversed. Consequently, even though this test was run for 12,000 timesteps (compared to $\sim$ 5,000 for the results presented in Chapter 4), the total simulation time is only around 12 ms (compared to $\sim$ 50 ms). It is likely that a bug is creating a new, large extrema in the velocity, which would limit the timestep across all cells. The nature of this bug must be investigated further, but we can still make some comparisons.

The BH-NS separation is given in Fig. 7.1, and the accretion rate is given in Fig. 7.2. The behaviour is qualitatively similar: the NS slowly inspirals towards the BH, there is a mass transfer episode, and the NS remnant is flung on a large, elliptical orbit. However, there are some clear quantitative differences. In the 'new' case, the first mass transfer episode happens sooner and at a larger separation distance. Furthermore, the elliptical orbit has a major axis around 20 km smaller. This could be explained by the approximate Riemann solver being more *diffuse*, an effect that was observed by e.g. Archibald (2008). This would cause the NS matter to 'spread out', and hence become disrupted by the BH at a larger separation distance. A consequence of this is that the peak accretion rate is smaller (0.027 $M_\odot$ ms^{-1} as opposed to 0.043 $M_\odot$ ms^{-1}). Additionally, the mass transfer appears to be much less stable.

The cumulative neutrino energy is given in Fig. 7.3. Recall that there is some unphysical neutrino emission for the first few ms, due to shock heating of the neutron star as it collides with the ambient background matter. The diffusive behaviour described above increases the

[1] Recall from Chapter 5 that the PPM and HLLD algorithms can handle the limit $\mathbf{B} \to 0$.

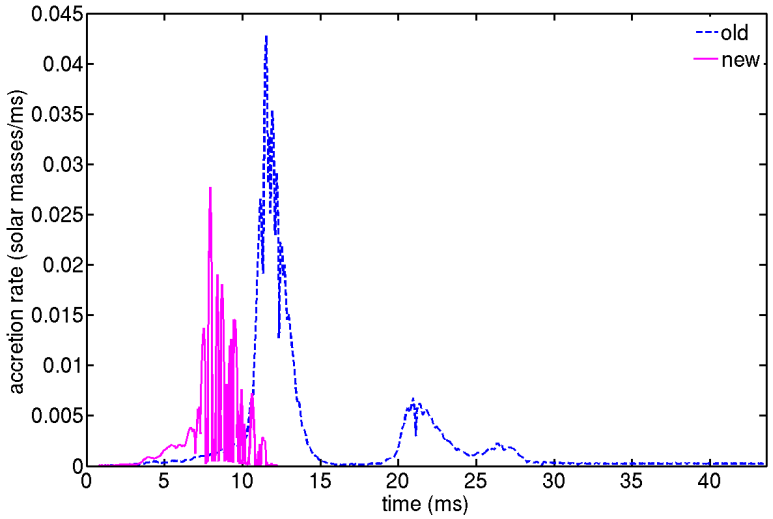


Figure 7.2: Accretion rate as a function of time, for the old code (blue/dotted) and the new version (magenta/solid) with all the implemented changes as described in the main text.

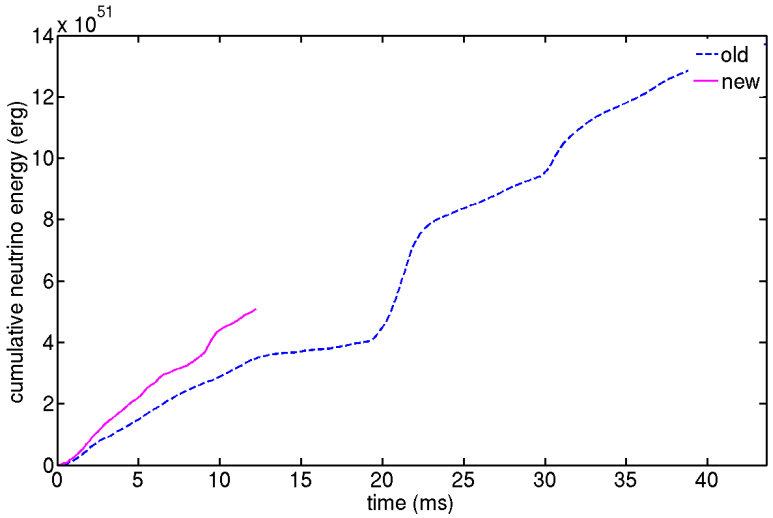


Figure 7.3: Cumulative neutrino energy as a function of time, for the old code (blue/dotted) and the new version (magenta/solid) with all the implemented changes as described in the main text.

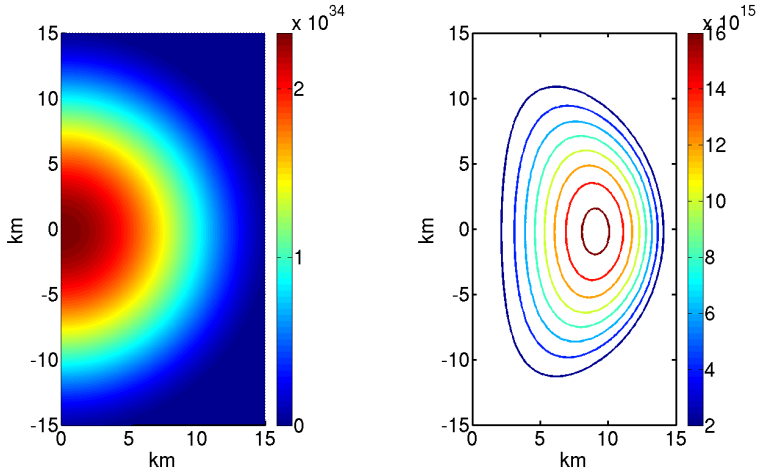


Figure 7.4: The magnetised neutron star. Left: the pressure as a function of radius. Right: the vector potential, A_ϕ. Note that the contours of the vector potential give the magnetic field lines.

amount of energy released through this process for the new case. However, at around 10 ms, the cumulative energy appears to grow at the same rate as for the original case.

Further study is required to determine the effect of the upgraded code on the late-time merger dynamics and neutrino emission, and the resulting disk mass.

7.2 Magnetised Neutron Stars

7.2.1 Setup

As an initial test for a magnetised BH-NS merger, we seed the initial neutron star with a weak poloidal field, following the method of Duez *et al.* (2006). Working in cylindrical polar co-ordinates (ω, ϕ, z), they define the following vector potential, A_ϕ:

$$A_\phi = \omega^2 \max[A_b(P - P_{\rm cut}), 0], \tag{7.1}$$

where A_b is a constant which determines the magnetic field strength, and the cutoff $P_{\rm cut}$ is set to 4% of the maximum pressure. A_b is chosen so that $\max(|\mathbf{B}|^2/p) \sim 10^{-2}$; we choose a value of $A_b = 2 \times 10^{20}$. This means that the gas pressure is much stronger than the magnetic pressure at early times, which makes this a relatively 'safe' initial test.

The magnetic field strength is then given as the curl of this potential. Since there is only one component, such that $\mathbf{A} = (0, A_\phi, 0)$, this is given by:

$$\mathbf{B} = \nabla \times \mathbf{A} = \left(-\frac{\partial A_\phi}{\partial z}\right)\hat{\omega} + \frac{1}{\omega}\left(\frac{\partial(\omega A_\phi)}{\partial \omega}\right)\hat{\mathbf{z}}, \tag{7.2}$$

which can then be converted into Cartesian co-ordinates. The magnetic field generated by this method is shown in Fig. 7.4, alongside the gas pressure for reference. Note that – although the magnetic pressure is much weaker than the gas pressure – the magnetic field still reaches

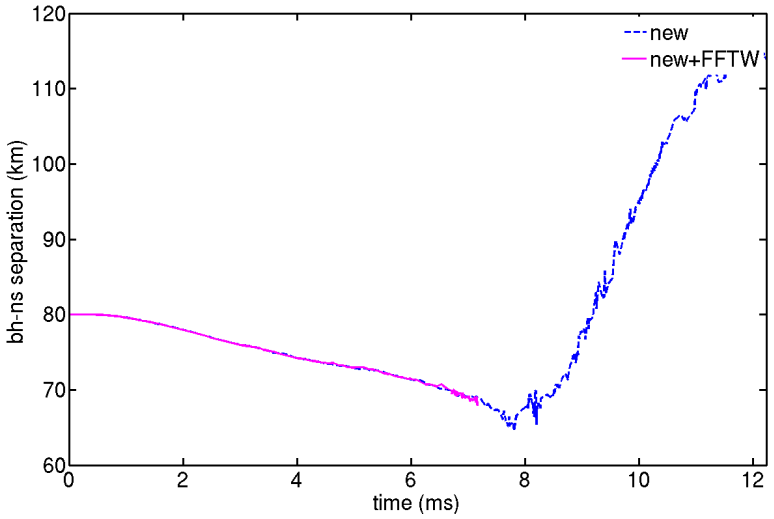


Figure 7.5: BH-NS separation as a function of time, for the new code (blue/dotted) and the new code with a full z-axis and FFTW library routines (magenta/solid)

strengths of $\sim 10^{15} - 10^{16}$ G. Recall from Section 1.2.2 that this is theoretically sufficient to power a GRB-like jet via the Blandford-Znajek mechanism (Lee *et al.*, 2000).

7.2.2 Further Issues

The magnetised neutron star model introduces two issues for Charybdis. The first is that the Cartesian components of the magnetic field are not symmetric about the orbital plane. Recall from Section 2.5 that this symmetry is assumed in the pure hydrodynamic case, and so Charybdis only models half of the z-axis. Hence, the code had to be expanded to model the entire z-axis, which required considerable developmental effort. The biggest change was that the Fast Fourier Transform routines – which were previously custom implemented to make use of this symmetry – were replaced with the FFTW library routines.[2] The $\mathbf{B} \rightarrow 0$ test was repeated, and the BH-NS separation is shown in Fig. 7.5. Note the simulated time has decreased further, as we have doubled the number of cells. However, these initial results suggest that this doubling and the FFTW routines were implemented correctly.

The second – and more significant issue – involves the black hole model. Charybdis is fundamentally Newtonian, and so the event horizon does not arise naturally in our simulation. Recall from Section 3.3.1 that we include it explicitly as a 'vacuum sphere': any mass that enters this sphere is removed from the grid and added to the BH point mass. This is a reasonable compromise, but creates a problem for magnetised models: what should happen to the magnetic field when it crosses the event horizon? It cannot be analgously deleted from the grid, as this will violate the divergence-free condition. Unfortunately, we did not have time to address this problem for inclusion in this thesis, and so it remains an open problem.

[2] The 'Fastest Fourier Transform in the West': `http://www.fftw.org`.

One possible approach would be to follow the work of Komissarov (2004), which uses the Kerr-Schild coordinate system. This allows another boundary for the computational domain to be placed *inside* the event horizon, where radiative boundary conditions could be imposed. However, this would require considerable restructuring of the existing code.

Chapter 8
Conclusions and Outlook

In this thesis, we have presented and developed numerical simulations of black hole-neutron star (BH-NS) mergers, to investigate whether they are plausible progenitors for the short-duration gamma-ray bursts. This work comprised two main projects. In the first project (Chapter 4), we investigated the effect of the equation of state on the merger dynamics. In the second project (Chapter 5-7), we began the process of upgrading our simulation code, Charybdis, to include magnetic field effects. This was done with the long-term goal of investigating magnetic processes during and after the merger. In this chapter, we summarise the key conclusions from this work, and offer thoughts for possible future directions.

8.1 Project One: the Equation of State

In Chapter 4, we presented the first BH-NS merger simulations to use the realistic EOSs of G. Shen *et al.* (2011b) and G. Shen *et al.* (2011a). For reference, we compared these with simulations that used the EOS of H. Shen *et al.* (1998a b), which was implemented previously. We found that the fate of the NS, the properties of the accretion disk and the neutrino emission were strongly dependent on the EOS. We also found that the two G. Shen EOSs produced sharp 'neutrino pulses', which were not observed in the H. Shen results. We believe this may be due to the more accurate physical models used to produce the two G. Shen EOSs.

We also observed 'orbital widening' of the NS after the first mass transfer event, which was previously thought to be suppressed by the use of GR-mimicking potentials. (We use the potential of Artemova *et al.*, 1996). We posit that this effect is dependent on the EOS, as well as other parameters such as the black hole mass and spin. Hence, a possible future project would be continue investigating the parameter space – with both pseudo-Newtonian and relativistic codes – in order to narrow down the true cause of this effect.

There are a number of possible routes for further investigation. Firstly, it would be useful to collect more information about the disk structure and temperature, to narrow down the exact cause of the aforementioned neutrino pulses. Also, for the viscosity model – described in Chapter 3 – we follow the work of Setiawan *et al.* (2006) and use an α value of $\alpha = 0.1$. However, it would be interesting to see how the properties of the disk change with this value. Finally, further EOSs have been released in the last two years, which have not yet been investigated in the context of BH-NS mergers. For example, the newer EOS of H. Shen *et al.*

(2011c) expands their previous EOS to include the effect of Λ hyperons. These are thought to appear in cold neutron star matter at high densities, and can soften the EOS. A number of EOSs have also been released by Hempel *et al.* (2012) and Steiner *et al.* (2013).

8.2 Project Two: Magnetic Fields

8.2.1 Modelling Magnetic Fields

In Chapter 5, we implemented four MHD reconstruction algorithms in a test code, Magneto, and presented 1D tests to compare them. Although none of these algorithms performed consistently better than the others in all metrics (e.g. L1-error in the density, number of oscillations, ability to capture sharp features), we found that the PPM reconstruction of Stone *et al.* (2008) will most likely be the best choice for our simulations of BH-NS mergers.

In Chapter 6, we presented a new algorithm for numerically solving the MHD equations in more than one dimension, while maintaining the divergence-free condition. This was based on the algorithm developed by Balsara and Spicer (1999) and Tóth (2000). Our algorithm has many advantages over the original version, including easier implementation in existing astrophysical codes, greater computational efficiency and better parallelisation. A possible future project would be to extend this algorithm to resistive MHD. This would be required to accurately model the magnetorotational instability in the disk, and magnetic braking effects (which could power a jet through the mechanism of Blandford and Znajek, 1977).

8.2.2 Towards Magnetised BH-NS Mergers

In Chapter 7, we presented preliminary tests which used these algorithms in simulations of BH-NS mergers (where $\mathbf{B} = 0$). We found that the upgraded code was more diffusive, which meant that the NS was disrupted sooner and at a larger separation distance. This could be due to the use of approximate Riemann solvers, which have been found to more diffuse than exact solvers (Archibald, 2008). We also found that the computational time modelled for a given number of timesteps was reduced considerably. The reason for this phenomena has not been determined, and hence requires further study.

We did not run any simulations of magnetised BH-NS mergers, due to an issue with our black hole model. Because our code is pseudo-Newtonian, we use a vacuum sphere to approximate the event horizon: any mass that enters this sphere is removed from the grid and added to the BH point mass. This is fine for pure hydrodynamics, but it is not clear how magnetic fields should be dealt with once they cross this sphere, without violating the divergence-free condition. One possible solution would be to extend the pseudo-Newtonian approximations – perhaps even to full relativity – so that the event horizon is handled naturally. For example, Komissarov (2004) use the Kerr-Schild coordinate system, which allows a new boundary to be placed inside the event horizon with radiative boundary conditions. However, we note that this would require substantial further development.

Bibliography

R. Archibald, *Numerical Simulations of Neutron Star Mergers as the Central Engines of Short-Period Gamma Ray Bursts*, Ph.D. thesis, University of Edinburgh (2008).

I. V. Artemova, G. Björnsson and I. D. Novikov, *Astrophys. J.*, **461** (1996), 565.

S. A. Balbus and J. F. Hawley, *Rev. Mod. Phys.*, **70** (1998), 1.

D. S. Balsara, *Astrophys. J. Suppl. S.*, **116** (1998), 133.

D. S. Balsara and D. S. Spicer, *J. Comput. Phys.*, **149** (1999), 270.

L. Blanchet, T. Damour and G. Schäfer, *Mon. Not. R. Astron. Soc.*, **242** (1990), 289.

R. D. Blandford and D. G. Payne, *Mon. Not. R. Astron. Soc.*, **199** (1982), 883.

R. D. Blandford and R. L. Znajek, *Mon. Not. R. Astron. Soc.*, **179** (1977), 433.

M. Brio and C. C. Wu, *J. Comput. Phys.*, **75** (1988), 400.

P. Colella, *J. Comput. Phys.*, **87** (1990), 171.

P. Colella and M. D. Sekora, *J. Comput. Phys.*, **227** (2008), 7069.

P. Colella and P. R. Woodward, *J. Comput. Phys.*, **54** (1984), 174.

E. Costa *et al.*, *Nature*, **387** (1997), 783.

W. Dai and P. R. Woodward, *J. Comput. Phys.*, **142** (1998), 331.

M. B. Deaton, M. D. Duez, F. Foucart, E. O'Connor, C. D. Ott, L. E. Kidder, C. D. Muhlberger, M. A. Scheel and B. Szilagyi, *Astrophys. J.*, **776** (2013), 47.

M. D. Duez, F. Foucart, L. E. Kidder, C. D. Ott and S. A. Teukolsky, *Class. Quantum. Grav.*, **27** (2010).

M. D. Duez, Y. T. Liu, S. L. Shapiro, M. Shibata and B. C. Stephens, *Phys. Rev. D*, **73** (2006), 104015.

D. Eichler, M. Livio, T. Piran and D. N. Schramm, *Nature*, **340** (1989), 126.

Z. B. Etienne, Y. T. Liu, V. Paschalidis and S. L. Shapiro, *Phys. Rev. D*, **85** (2012), 064029.

C. R. Evans and J. F. Hawley, *Astrophys. J.*, **332** (1988), 659.

J. A. Faber and F. A. Rasio, *Living Rev. Relativity*, **15** (2012), 8.

S. A. E. G. Falle and S. S. Komissarov, *J. Plasma Physics*, **65** (2001), 29.

G. J. Fishman and C. A. Meegan, *Annu. Rev. Astron. Astrophys.*, **33** (1995), 415.

T. A. Gardiner and J. M. Stone, *J. Comput. Phys.*, **205** (2005), 509.

N. Gehrels *et al.*, *Nature*, **437** (2005), 851.

S. K. Godunov, *Math. Sbornik*, **47** (1959), 231.

A. Harten, P. D. Lax and B. V. Leer, *SIAM Review*, **25** (1983), 35.

M. Hempel, T. Fischer, J. Schaffner-Bielich and M. Liebendörfer, *Astrophys. J.*, **748** (2012), 70.

R. A. Hulse and J. H. Taylor, *Astrophys. J*, **194** (1975), L51.

P. Janhunen, *J. Comput. Phys.*, **160** (2000), 649 .

H.-T. Janka, T. Eberl, M. Ruffert and C. L. Fryer, *Astrophys. J.*, **527** (1999), L39.

V. Kalogera *et al.*, *Astrophys. J.*, **601** (2004), L179.

R. W. Klebesadel, I. B. Strong and R. A. Olson, *Astrophys. J.*, **182** (1973), L85.

S. S. Komissarov, *Mon. Not. R. Astron. Soc*, **350** (2004), 1431.

J. M. Lattimer and F. D. Swesty, *Nucl. Phys. A.*, **535** (1991).

H. K. Lee, R. A. M. J. Wijers and G. E. Brown, *Phys. Rep.*, **325** (2000), 83.

W. H. Lee, E. Ramirez-Ruiz and D. Page, *Astrophys. J.*, **632** (2005), 421.

W.-H. Lei, B. Zhang and E.-W. Liang, *Astrophys. J.*, **765** (2013), 125 .

A. J. Levan *et al.*, *Mon. Not. R. Astron. Soc.*, **378** (2007), 1439.

A. J. Levan *et al.*, *Astrophys. J.*, **781** (2014), 13.

R. J. LeVeque, *Numerical Methods for Conservation Laws* (Birkhäuser, 1992).

R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems* (Cambridge University Press, 2002).

A. Levinson and D. Eichler, *Phys. Rev. Lett.*, **85** (2000), 236.

S. Li, *J. Comput. Phys.*, **203** (2005), 344.

R. Liska and B. Wendroff, *SIAM. J. Sci. Comput.*, **25** (2003), 995.

P. Londrillo and L. del Zanna, *Astrophys. J.*, **530** (2000).

M. C. Miller, *Astrophys. J*, **626** (2005), L41.

T. Miyoshi and K. Kusano, *J. Comput. Phys.*, **208** (2005), 315.

E. Nakar, *Phys. Rep.*, **442** (2007), 166.

R. Narayan, B. Paczynski and T. Piran, *Astrophys. J.*, **395** (1992), L83.

R. J. Nemiroff, *Comments Astrophys.*, **17** (1994), 189.

R. Oechslin, H.-T. Janka and A. Marek, *A & A*, **467** (2007), 395.

S. A. Orszag and C.-M. Tang, *J. Fluid. Mech.*, **90** (1979), 129 .

B. Paczyński and P. Wiita, *Astron. Astrophys.*, **88** (1980), 23.

J. van Paradijs *et al.*, *Nature*, **386** (1997), 686.

D. J. Price and S. Rosswog, *Science*, **312** (2006), 719 .

M. J. Rees and P. Mészáros, *Astrophys. J.*, **430** (1994), L93.

L. Rezzolla, B. Giacomazzo, L. Baiotti, J. Granot, C. Kouveliotou and M. A. Aloy, *Astrophys. J.*, **632** (2011), L6.

S. Rosswog, *Astrophys. J.*, **634** (2005), 1202.

S. Rosswog, in 'Proceedings of Science (NIC XI) 032', (2010).

S. Rosswog and M. B. Davies, *Mon. Not. R. Astron. Soc*, **334** (2002), 481.

S. Rosswog, R. Speith and G. A. Wynn, *Mon. Not. R. Astron. Soc.*, **351** (2004), 1121.

M. Ruffert, *Astron. Astrophys.*, **265** (1992), 82.

M. Ruffert and H.-T. Janka, **380** (2001), 544.

M. Ruffert and H.-T. Janka, *Astron. Astrophys.*, **514** (2010), A66.

M. Ruffert, H.-T. Janka and G. Schäfer, *Astron. Astrophys.*, **311** (1996), 532.

D. Ryu and T. W. Jones, *Astrophys. J.*, **442** (1995), 228.

R. Sari, T. Piran and R. Narayan, *Astrophys. J.*, **497** (1998), L17.

S. Setiawan, M. Ruffert and H.-T. Janka, *Astron. Astrophys.*, **458** (2006), 553.

N. I. Shakura and R. A. Sunyaev, *Astron. Astrophys.*, **24** (1973), 337.

G. Shen, C. J. Horowitz and E. O'Connor, *Phys. Rev. C.*, **83** (2011a), 065808.

G. Shen, C. J. Horowitz and S. Teige, *Phys. Rev. C*, **82** (2010), 045802.

G. Shen, C. J. Horowitz and S. Teige, *Phys. Rev. C*, **83** (2011b), 035802.

H. Shen, H. Toki, K. Oyamatsu and K. Sumiyoshi, *Nucl. Phys. A*, **637** (1998a), 435.

H. Shen, H. Toki, K. Oyamatsu and K. Sumiyoshi, *Prog. Theor. Phys.*, **100** (1998b), 1013 .

H. Shen, H. Toki, K. Oyamatsu and K. Sumiyoshi, *Astrophys. J. Suppl. S.*, **197** (2011c), 20.

M. Shibata and Y. Sekiguchi, *Prog. Theor. Phys.*, **127** (2012), 535.

M. Shibata and K. Taniguchi, *Living Rev. Relativity*, **14** (2011), 6.

H. Sriskantha and M. Ruffert, *Mon. Not. Royal. Astro. Soc.*, **432** (2013), 2579.

A. W. Steiner, M. Hempel and T. Fischer, *Astrophys. J.*, **774** (2013), 17.

J. M. Stone, T. A. Gardiner, J. F. Hawley and J. . B. Simon, *Astrophys. J. Suppl. S.*, **178** (2008), 137.

E. F. Toro, M. Spruce and W. Speares, *Shock Waves*, **4** (1994), 25.

G. Tóth, *J. Comput. Phys.*, **161** (2000), 605.

B. Zhang, *C. R. Physique*, **12** (2011), 206.

Appendix A
The MHD Equations

The 1D MHD Equations

For the work on magnetic fields presented in this thesis, we use the component form of the equations of magnetohydrodynamics (MHD), given by e.g. Stone *et al.* (2008):

$$\frac{\partial}{\partial t}\begin{bmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ \mathcal{E} \\ B_y \\ B_z \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} \rho v_x \\ \rho v_x^2 + p_T - B_x^2 \\ \rho v_x v_y - B_x B_y \\ \rho v_x v_z - B_x B_z \\ v_x(\mathcal{E} + p_T) - B_x(\mathbf{v}\cdot\mathbf{B}) \\ -E_z = v_x B_y - v_y B_x \\ E_y = v_x B_z - v_z B_x \end{bmatrix} = 0, \tag{A.1}$$

where ρ is the mass density, $(\rho v_x, \rho v_y, \rho v_z)$ are the components of the linear momentum, $\mathcal{E}$ is the total energy, (B_x, B_y, B_z) are the components of the magnetic field and p_T is the total pressure. Further details about this form of the MHD equations is given in Section 5.1.1.

The Flux Jacobian Matrix

The above equations are in the form $\partial\mathbf{u}/\partial t + \partial\mathbf{f}/\partial x = 0$. For the reconstruction algorithms presented in Chapter 5, however, we work in the primitive variables $\mathbf{v}$:

$$\mathbf{v} = [\rho, v_x, v_y, v_z, p_G, B_y, B_z], \tag{A.2}$$

where (v_x, v_y, v_z) are the components of the velocity and p_G is the gas pressure. The flux Jacobian matrix, $\mathbf{A} = \partial\mathbf{F}/\partial\mathbf{v}$, is then given by eq. A8 of Stone *et al.* (2008):

$$\mathbf{A} = \begin{bmatrix} v_x & \rho & 0 & 0 & 0 & 0 & 0 \\ 0 & v_x & 0 & 0 & 1/\rho & B_y/\rho & B_z/\rho \\ 0 & 0 & v_x & 0 & 0 & -B_x/\rho & 0 \\ 0 & 0 & 0 & v_x & 0 & 0 & -B_x/\rho \\ 0 & \rho a^2 & 0 & 0 & v_x & 0 & 0 \\ 0 & B_y & -B_x & 0 & 0 & v_x & 0 \\ 0 & B_z & 0 & -B_x & 0 & 0 & v_x \end{bmatrix}, \tag{A.3}$$

where $a^2 = \gamma p_G/\rho$ is the sound speed and γ is the adiabatic index.

The Eigenvalues

The seven eigenvalues are given by eqs. A9-A11 of Stone *et al.* (2008):

$$\lambda = (v_x - C_f,\ v_x - C_{Ax},\ v_x - C_s,\ v_x,\ v_x + C_s,\ v_x + C_{Ax},\ v_x + C_f), \tag{A.4}$$

where the corresponding wavespeeds are given by:

$$C^2_{f,s} = \frac{1}{2}\left[(a^2 + C_A^2) \pm \sqrt{(a^2 + C_A^2)^2 - 4a^2 C^2_{Ax}}\right], \tag{A.5}$$

$$C_A^2 = \left(B_x^2 + B_y^2 + B_z^2\right)/\rho, \tag{A.6}$$

$$C^2_{Ax} = B_x^2/\rho. \tag{A.7}$$

The Eigenvectors

In order to write down the eigensystem, Stone *et al.* (2008) use the following terms:

$$C_{ff} = C_f \alpha_f, \qquad C_{ss} = C_s \alpha_s, \tag{A.8}$$

$$Q_f = C_f \alpha_f S, \qquad Q_s = C_s \alpha_s S, \tag{A.9}$$

$$A_f = a\alpha_f \sqrt{\rho}, \qquad A_s = a\alpha_s \sqrt{\rho}, \tag{A.10}$$

$$\alpha_f^2 = \frac{a^2 - C_s^2}{C_f^2 - C_s^2}, \qquad \alpha_s^2 = \frac{C_f^2 - a^2}{C_f^2 - C_s^2}, \tag{A.11}$$

$$\beta_y = \frac{B_y}{\sqrt{B_y^2 + B_z^2}}, \qquad \beta_z = \frac{B_z}{\sqrt{B_y^2 + B_z^2}} \tag{A.12}$$

where $S = \mathrm{sgn}(B_x)$. There are a couple of special cases to consider: (a) if $C_A = C_{Ax} = a$ (and hence $C_f = C_s$), then $\alpha_f = 1$ and $\alpha_s = 0$; (b) if $B_y = B_z = 0$, then $\beta_y = \beta_z = 0$.

The right eigenvectors are the columns of the matrix (Stone *et al.*, 2008, eq. A12):

$$\mathbf{R} = \begin{bmatrix} \rho\alpha_f & 0 & \rho\alpha_s & 1 & \rho\alpha_s & 0 & \rho\alpha_f \\ -C_{ff} & 0 & -C_{ss} & 0 & C_{ss} & 0 & C_{ff} \\ Q_s\beta_y & -\beta_z & -Q_f\beta_y & 0 & Q_f\beta_y & \beta_z & -Q_s\beta_y \\ Q_s\beta_z & \beta_y & -Q_f\beta_z & 0 & Q_f\beta_z & -\beta_y & -Q_s\beta_z \\ \rho a^2\alpha_f & 0 & \rho a^2\alpha_s & 0 & \rho a^2\alpha_s & 0 & \rho a^2\alpha_f \\ A_s\beta_y & -\beta_z S\sqrt{\rho} & -A_f\beta_y & 0 & -A_f\beta_y & -\beta_z S\sqrt{\rho} & A_s\beta_y \\ A_s\beta_z & \beta_y S\sqrt{\rho} & -A_f\beta_z & 0 & -A_f\beta_z & \beta_y S\sqrt{\rho} & A_s\beta_z \end{bmatrix}, \tag{A.13}$$

while the left eigenvectors are the rows of the matrix (Stone *et al.*, 2008, eq. A18):

$$\mathbf{L} = \begin{bmatrix} 0 & -N_f C_{ff} & N_f Q_s\beta_y & N_f Q_s\beta_z & N_f\alpha_f/\rho & N_f A_s\beta_y/\rho & N_f A_s\beta_z/\rho \\ 0 & 0 & -\beta_z/2 & \beta_y/2 & 0 & -\beta_z S/(2\sqrt{\rho}) & \beta_y S/(2\sqrt{\rho}) \\ 0 & -N_s C_{ss} & -N_s Q_f\beta_y & -N_s Q_f\beta_z & N_s\alpha_s/\rho & -N_s A_f\beta_y/\rho & -N_s A_f\beta_z/\rho \\ 1 & 0 & 0 & 0 & -1/a^2 & 0 & 0 \\ 0 & N_s C_{ss} & N_s Q_f\beta_y & N_s Q_f\beta_z & N_s\alpha_s/\rho & -N_s A_f\beta_y/\rho & -N_s A_f\beta_z/\rho \\ 0 & 0 & \beta_z/2 & -\beta_y/2 & 0 & -\beta_z S/(2\sqrt{\rho}) & \beta_y S/(2\sqrt{\rho}) \\ 0 & N_f C_{ff} & -N_f Q_s\beta_y & -N_f Q_s\beta_z & N_f\alpha_f/\rho & N_f A_s\beta_y/\rho & N_f A_s\beta_z/\rho \end{bmatrix},$$

where $N_f = N_s = 1/(2a^2)$ are normalisation factors.

For reference, these eigensystem calculations have been implemented in subroutine `[REC07]` of our test code, Magneto. See Section 5.1.2 for more information.

Appendix B
Expanded Derivations for the Constrained Transport Algorithm

B.1 The Cell-Averaged Magnetic Field

In this section, we expand the derivation of eq. (6.16), which encapsulates the original Balsara-Spicer-Tóth algorithm. The first line of eq. (6.16) gives:

$$(Bx)^{\bullet}_{n+1} = \frac{1}{2}\left[(bx)^{i*}_{n+1} + (bx)^{i*-1}_{n+1}\right].$$

Using eq. (6.6), we can substitute for the (bx) terms:

$$2(Bx)^{\bullet}_{n+1} = (bx)^{i*}_{n} - \frac{\Delta t}{\Delta y}\left[(Ez)^{i*,j*}_{n*} - (Ez)^{i*,j*-1}_{n*}\right] + \frac{\Delta t}{\Delta z}\left[(Ey)^{i*,k*}_{n*} - (Ey)^{i*,k*-1}_{n*}\right]$$

$$+ (bx)^{i*-1}_{n} - \frac{\Delta t}{\Delta y}\left[(Ez)^{i*-1,j*}_{n*} - (Ez)^{i*-1,j*-1}_{n*}\right] + \frac{\Delta t}{\Delta z}\left[(Ey)^{i*-1,k*}_{n*} - (Ey)^{i*-1,k*-1}_{n*}\right],$$

where we have moved the factor of 2 to the LHS. We can regroup these as:

$$\begin{aligned}(Bx)^{\bullet}_{n+1} = \frac{1}{2}&\left[(bx)^{i*}_{n} + (bx)^{i*-1}_{n}\right] \\ &- \frac{\Delta t}{\Delta y}\frac{1}{2}\left[(Ez)^{i*,j*}_{n*} + (Ez)^{i*-1,j*}_{n*}\right] \\ &+ \frac{\Delta t}{\Delta y}\frac{1}{2}\left[(Ez)^{i*,j*-1}_{n*} + (Ez)^{i*-1,j*-1}_{n*}\right] \\ &+ \frac{\Delta t}{\Delta z}\frac{1}{2}\left[(Ey)^{i*,k*}_{n*} + (Ey)^{i*-1,k*}_{n*}\right] \\ &- \frac{\Delta t}{\Delta z}\frac{1}{2}\left[(Ey)^{i*,k*-1}_{n*} + (Ey)^{i*-1,k*-1}_{n*}\right].\end{aligned} \tag{B.1}$$

We then define the following interpolations of the edge-averaged electric field components:

$$(Ez)^{j*}_{\to y} = \frac{1}{2}\left[(Ez)^{i*,j*}_{n*} + (Ez)^{i*-1,j*}_{n*}\right], \tag{B.2}$$

$$(Ey)^{k*}_{\to z} = \frac{1}{2}\left[(Ey)^{i*,k*}_{n*} + (Ey)^{i*-1,k*}_{n*}\right]. \tag{B.3}$$

If we substitute eqs. (A.2) and (A.3) into eq. (A.1), we retrieve the second line of eq. (6.16), as required. Eqs. (6.17) and (6.18) can be found in a similar manner.

B.2 A Modified Approach

We now expand the derivation of eq. (6.27). The expressions for the electric field components in eq. (6.16), in terms of fluxes returned by the Riemann solver, are given in eqs. (6.21) - (6.26). If we substitute these in, then we get the expression:

$$
\begin{aligned}
(Bx)^{\bullet}_{n+1} = (Bx)^{\bullet}_{n} \\
&- \frac{\Delta t}{\Delta y}\frac{1}{8}\left[(\tilde{G6})^{i,j*}_{n*} + (\tilde{G6})^{i+1,j*}_{n*} - (\tilde{F7})^{i*,j}_{n*} - (\tilde{F7})^{i*,j+1}_{n*}\right] \\
&- \frac{\Delta t}{\Delta y}\frac{1}{8}\left[(\tilde{G6})^{i-1,j*}_{n*} + (\tilde{G6})^{i,j*}_{n*} - (\tilde{F7})^{i*-1,j}_{n*} - (\tilde{F7})^{i*-1,j+1}_{n*}\right] \\
&+ \frac{\Delta t}{\Delta y}\frac{1}{8}\left[(\tilde{G6})^{i,j*-1}_{n*} + (\tilde{G6})^{i+1,j*-1}_{n*} - (\tilde{F7})^{i*,j-1}_{n*} - (\tilde{F7})^{i*,j}_{n*}\right] \\
&+ \frac{\Delta t}{\Delta y}\frac{1}{8}\left[(\tilde{G6})^{i-1,j*-1}_{n*} + (\tilde{G6})^{i,j*-1}_{n*} - (\tilde{F7})^{i*-1,j-1}_{n*} - (\tilde{F7})^{i*-1,j}_{n*}\right] \\
&+ \frac{\Delta t}{\Delta z}\frac{1}{8}\left[(\tilde{F8})^{i*,k}_{n*} + (\tilde{F8})^{i*,k+1}_{n*} - (\tilde{H6})^{i,k*}_{n*} - (\tilde{H6})^{i+1,k*}_{n*}\right] \\
&+ \frac{\Delta t}{\Delta z}\frac{1}{8}\left[(\tilde{F8})^{i*-1,k}_{n*} + (\tilde{F8})^{i*-1,k+1}_{n*} - (\tilde{H6})^{i-1,k*}_{n*} - (\tilde{H6})^{i,k*}_{n*}\right] \\
&+ \frac{\Delta t}{\Delta z}\frac{1}{8}\left[(\tilde{F8})^{i*,k-1}_{n*} + (\tilde{F8})^{i*,k}_{n*} - (\tilde{H6})^{i,k*-1}_{n*} - (\tilde{H6})^{i+1,k*-1}_{n*}\right] \\
&+ \frac{\Delta t}{\Delta z}\frac{1}{8}\left[(\tilde{F8})^{i*-1,k-1}_{n*} + (\tilde{F8})^{i*-1,k}_{n*} - (\tilde{H6})^{i-1,k*-1}_{n*} - (\tilde{H6})^{i,k*-1}_{n*}\right].
\end{aligned}
$$

However, in eq. (6.27), we are only interested in the contributions to this expression from the x-direction pass. These are the fluxes F, highlighted in green. If we consider only these terms, and simplify them, then we obtain eq. (6.27).

Appendix C
Published Work

The work described in Chapter 6 has been published (Sriskantha and Ruffert, 2013). For reference, a copy of the submitted paper has been included here.

Mon. Not. R. Astron. Soc. **000**, 1–12 (2013) Printed 27 March 2014 (MN LATEX style file v2.2)

A partially dimensionally-split approach to numerical MHD

H. Sriskantha⋆, M. Ruffert
Maxwell Institute for Mathematical Sciences, School of Mathematics, The University of Edinburgh, Edinburgh, EH9 3JZ, UK.

Written Nov 2012 - Mar 2013.

ABSTRACT
We modify an existing magnetohydrodynamics algorithm to make it more compatible with a dimensionally-split (DS) framework. It is based on the standard reconstruct-solve-average strategy (using a Riemann solver), and relies on constrained transport to ensure that the magnetic field remains divergence-free ($\nabla \cdot \boldsymbol{B} = 0$). The DS approach, combined with the use of a single, cell-centred grid (for both the fluid quantities and the magnetic field), means that the algorithm can be easily added to existing DS hydrodynamics codes. This makes it particularly useful for mature astrophysical codes, which often model more complicated physical effects on top of an underlying DS hydrodynamics engine, and therefore cannot be restructured easily. Several test problems have been included to demonstrate the accuracy of the algorithm, and illustrative source code has been made freely available online.

Key words: MHD – methods: numerical – magnetic fields

1 INTRODUCTION

When developing complex astrophysical codes, we often start with a simple hydrodynamics engine. This is designed to numerically solve the Euler equations of pure hydrodynamics, which – in conservative form – can be written as:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial y} + \frac{\partial \boldsymbol{H}}{\partial z} = 0, \tag{1}$$

where $\boldsymbol{U}$ is the vector of state quantities we wish to evolve in time, and the flux vectors $\boldsymbol{F}(\boldsymbol{U})$, $\boldsymbol{G}(\boldsymbol{U})$ and $\boldsymbol{H}(\boldsymbol{U})$ contain their respective fluxes in the three spatial (Cartesian) dimensions. A popular approach for solving these equations numerically is the reconstruct-solve-average (RSA) strategy (a term coined by Mignone et al. 2007). We start by dividing our space into a 3D grids of 'cells', and then we discretise the initial data by storing only the cell-volume averages of the state quantities at every cell centre. Each timestep then comprises three stages. First, we *reconstruct* this discretised data to estimate the value of the state quantities on either side of each cell face, thus defining – in effect – a series of 'Riemann problems'.[1] Next, we *solve* each of these problems independently using a Riemann solver algorithm, which determines the fluxes of the state quantities through each cell face. Finally, we use these inter-cell fluxes to determine the new cell-averaged state quantities. These three stages are repeated until the desired amount of 'time' has been simulated.

This strategy is often combined with dimensional splitting (DS), where the 3D grid is split into several independent 1D problems and each timestep then becomes a series of 'directional passes'. In each of these passes, all of the rows in a given direction are evolved according to the RSA strategy above. However, we only deal with the cell faces which are perpendicular to the direction of the pass: the x-direction pass determines the flux vector $\boldsymbol{F}(\boldsymbol{U})$ through all the $x =$ constant faces, while the y- and z- direction passes determine $\boldsymbol{G}(\boldsymbol{U})$ and $\boldsymbol{H}(\boldsymbol{U})$ through the $y =$ constant and $z =$ constant faces respectively. The popularity of this approach is due to its many advantages: the timestep is not as constricted by stability conditions as with a fully multi-dimensional code; it is simple to upgrade an existing 1D code to more dimensions in this way; the rows in each pass can be evolved independently, which means the code can be easily parallelised; it has been shown that the accuracy of the 1D code can be preserved by alternating the order of the passes (Liska & Wendroff 2003).

Once this hydrodynamics engine is in place, it can be improved by adding other physical phenomena (such as viscosity, self-gravity or a realistic equation of state), and more sophisticated computational features (such as parallelisation and adaptive/non-uniform grids). This is generally an effective way to develop code, but can cause difficulties if we then need to modify the underlying engine – for example, when

⋆ E-mail: hari.sriskantha@ed.ac.uk
[1] A Riemann problem is defined as the solution to a conservation law with piecewise constant initial data and a single discontinuity (LeVeque 1992). Algorithms exist to find this solution both exactly and approximately (the latter sacrificing some accuracy in favour of computational speed).

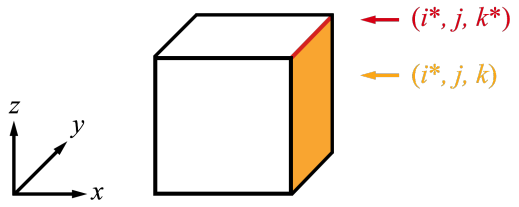


Figure 1. This figure depicts the cell centred at position (i, j, k). The orange face is labelled $(i*, j, k)$, where $i* = i + 1/2$, because it is located $\frac{1}{2}\Delta x$ away from the cell centre. Using similar reasoning, the red edge is labelled $(i*, j, k*)$. (See §2.1 for details.)

adding the effects of magnetic fields. The dynamics of a fluid in a magnetic field are governed by the equations of magnetohydrodynamics (MHD), which combine the Euler equations with Maxwell's equations of electromagnetics. Although these equations can be written in the same form as eq. (1), they cannot be easily incorporated into DS-RSA codes because of the need to maintain the divergence-free condition:

$$\nabla \cdot \boldsymbol{B} = 0. \tag{2}$$

Ignoring this condition in numerical simulations can result in unwanted effects, such as the creation of unphysical forces parallel to the magnetic field, and the loss of conservation of momentum and energy (Balsara & Spicer 1999).

There are several families of algorithms for maintaining this condition (which are summarised in e.g. Tóth 2000). In this paper, we focus on the constrained transport (CT) family, where the magnetic field quantities are discretised in such a way that $\nabla \cdot \boldsymbol{B}$ does not change in time analytically (to machine error). We then simply choose initial conditions such that $\nabla \cdot \boldsymbol{B} = 0$. Many CT algorithms have been developed (such as Dai & Woodward 1998; Balsara & Spicer 1999; Tóth 2000; Londrillo & del Zanna 2000), but these are often dimensionally unsplit and/or require a 'staggered' grid: the fluid quantities are stored as volume-averages at cell centres, while the magnetic field quantities are stored as *surface*-averages at cell *faces*. This is not troublesome for researchers intending to develop new codes from scratch, but is less than ideal for those working on mature codes as described above. Even implementing the non-staggered algorithm suggested in Tóth (2000) – which was modified from the algorithm of Balsara & Spicer (1999) – would require considerable restructuring of the underlying hydrodynamics engine. Instead, a preferable algorithm is one that minimises the need for such restructuring. In this paper, we show that one can be developed by reframing Tóth's approach so that it is more compatible with a dimensionally-split framework. By doing so, we aim to considerably simplify the development required to add the MHD equations to mature astrophysical codes.

In §2 we describe some practical details, including notation, units and the formulation of the MHD equations used. In §3 we summarise the Balsara-Spicer-Tóth algorithm for maintaining the divergence-free condition, while in §4 we derive our 'more' dimensionally-split version. Readers who are only interested in the final algorithm may skip ahead to §5, where we include a clear step-by-step guide to implementing it. (Additionally, illustrative source code has been made available online – which was designed to be as easy as possible to be read – in order to aid understanding of the algorithm.) Finally, in §6 we demonstrate its accuracy with several well-known MHD tests.

2 FORMULATION AND NOTATION

2.1 Discretisation

We model Cartesian space in the domain $D_x \times D_y \times D_z$. We divide this space into $n_x \times n_y \times n_z$ 'cells' in the usual way, each of volume $\Delta x \times \Delta y \times \Delta z$, such that $D_x = n_x \cdot \Delta x$, etc. (We assume that the cell sizes are constant and uniform, but it is simple to extend our algorithm to adaptive/non-uniform grids.) The position of each cell centre is labelled as (i, j, k), where $i \in \{1, 2, ..., n_x\}$, etc. We also refer to the faces and edges of cells, using starred notation $i* = i + 1/2$. A face will therefore have one starred co-ordinate, e.g. $(i*, j, k)$, while an edge will have two, e.g. $(i*, j*, k)$. A typical cell, face and edge are depicted in Fig. 1. Finally, we discretise time such that $t_n = n \cdot \Delta t_n$. The timestep Δt_n can change with time, but must always satisfy the Courant-Friedrich-Lewy (CFL) stability condition.

To avoid excessive indices, we only write down spatial co-ordinates that *differ* from (i, j, k). For example, the face-average $\boldsymbol{U}_n^{i*,j,k+1}$ will be abbrievated to $\boldsymbol{U}_n^{i*,k+1}$. The reader should therefore assume that there are always three spatial co-ordinates, even if all three are not explicitly written down. When there is no change from (i, j, k) we use $^\bullet$, such that $\boldsymbol{U}_n^{i,j,k}$ is abbrievated to $\boldsymbol{U}_n^\bullet$. Similarly, we do not use subscripts to denote components of discretised values. For example: the cell average of the x-component of the magnetic field will be written as $(Bx)_n^\bullet$, not $B_{x,n}^\bullet$. A summary of the non-standard notation used throughout this paper is provided in Table 1.

Using the discretisation above, we can now write a numerical equation for the DS-RSA strategy, which gives $\boldsymbol{U}_{n+1}^\bullet$ in terms of $\boldsymbol{U}_n^\bullet$:

$$\boldsymbol{U}_{n+1}^\bullet = \boldsymbol{U}_n^\bullet - \frac{\Delta t}{\Delta x}(\tilde{\boldsymbol{F}}_{n*}^{i*} - \tilde{\boldsymbol{F}}_{n*}^{i*-1}) - \frac{\Delta t}{\Delta y}(\tilde{\boldsymbol{G}}_{n*}^{j*} - \tilde{\boldsymbol{G}}_{n*}^{j*-1}) - \frac{\Delta t}{\Delta z}(\tilde{\boldsymbol{H}}_{n*}^{k*} - \tilde{\boldsymbol{H}}_{n*}^{k*-1}), \tag{3}$$

Table 1. A summary of the non-standard notation used throughout this paper. See §2.1 for more details.

Notation	Meaning
$\bullet$	the cell centred at (i, j, k)
$(j+1)$	the cell centred at $(i, j+1, k)$
$i*$	the index $i + 1/2$
$(i*, k+1)$	the cell face centred at $(i*, j, k+1)$
$\tilde{\boldsymbol{F}}, \tilde{\boldsymbol{G}}, \tilde{\boldsymbol{H}}$	inter-cell flux vectors returned by the Riemann solver
$(\tilde{F}7)$	the 7th component of $\tilde{\boldsymbol{F}}$
(Bx)	the x-component of $\boldsymbol{B}$
φ	$\varphi = 1$ or $1/\sqrt{4\pi}$ depending on units (see §2.2)

where the last three terms correspond respectively to the three directional passes. The terms $\tilde{\boldsymbol{F}}^{i*}_{n*}$, $\tilde{\boldsymbol{G}}^{j*}_{n*}$ and $\tilde{\boldsymbol{H}}^{k*}_{n*}$ are the inter-cell flux vectors returned by the Riemann solver during the 'solve' stage (signified by tildes), and $n* = n + 1/2$ refers to the half-timestep.

2.2 Units

In the literature, there are two sets of units commonly used for the magnetic field, which differ by a factor of $\sqrt{4\pi}$. To aid compatibility, we use unit-neutral notation, using a constant φ. For SI units, $\varphi = 1$, while for CGS units, $\varphi = 1/\sqrt{4\pi}$. By including this constant in the code, we can switch easily between the two. We will make it clear which version is used by any referenced papers.

2.3 The MHD equations

The MHD equations can be written in the same form as eq. (1). The vector of state quantities is defined as:

$$\boldsymbol{U} = [\rho, \rho v_x, \rho v_y, \rho v_z, \mathcal{E}, B_x, B_y, B_z]^T, \tag{4}$$

where ρ is the mass density, $\rho\boldsymbol{v}$ is the momentum density, $\mathcal{E}$ is the total energy density, and $\boldsymbol{B}$ is the magnetic field density. Recall that the cell-averaged values of these quantities are stored at every cell centre. We refer to the first five elements as 'fluid quantities' and the last three elements as 'magnetic field quantities' whenever we need to distinguish between the two. The flux vectors are defined as:

$$\boldsymbol{F} = \begin{bmatrix} \rho v_x \\ \rho v_x^2 + p_\mathrm{T} - \varphi^2 B_x^2 \\ \rho v_x v_y - \varphi^2 B_x B_y \\ \rho v_x v_z - \varphi^2 B_x B_z \\ v_x(\mathcal{E} + p_\mathrm{T}) - \varphi^2 B_x(\boldsymbol{v}\cdot\boldsymbol{B}) \\ 0 \\ -E_z \\ E_y \end{bmatrix}, \quad \boldsymbol{G} = \begin{bmatrix} \rho v_y \\ \rho v_x u_y - \varphi^2 B_x B_y \\ \rho v_y^2 + p_\mathrm{T} - \varphi^2 B_y^2 \\ \rho v_y v_z - \varphi^2 B_y B_z \\ v_y(\mathcal{E} + p_\mathrm{T}) - \varphi^2 B_y(\boldsymbol{v}\cdot\boldsymbol{B}) \\ E_z \\ 0 \\ -E_x \end{bmatrix}, \quad \boldsymbol{H} = \begin{bmatrix} \rho v_z \\ \rho v_x v_z - \varphi^2 B_x B_z \\ \rho v_y v_z - \varphi^2 B_y B_z \\ \rho v_z^2 + p_\mathrm{T} - \varphi^2 B_z^2 \\ v_z(\mathcal{E} + p_\mathrm{T}) - \varphi^2 B_z(\boldsymbol{v}\cdot\boldsymbol{B}) \\ -E_y \\ E_x \\ 0 \end{bmatrix}, \tag{5}$$

where p_T is the total pressure, such that $p_\mathrm{T} = p_\mathrm{M} + p_\mathrm{G}$. The magnetic pressure is given by $p_\mathrm{M} = \frac{1}{2}\varphi^2 B^2$, where $B^2 = \boldsymbol{B}\cdot\boldsymbol{B}$. The gas pressure, p_G, is related to the total energy density by an ideal equation of state, which closes the system, such that:

$$\mathcal{E} = \frac{p_\mathrm{G}}{\gamma - 1} + \frac{1}{2}\rho\boldsymbol{v}^2 + \frac{1}{2}\varphi^2 B^2, \tag{6}$$

where γ is the ratio of specific heats. Finally, $\boldsymbol{E}$ is the electric field, which is given by the ideal version of Ohm's law:

$$\boldsymbol{E} = -\boldsymbol{v}\times\boldsymbol{B}. \tag{7}$$

3 BALSARA-SPICER-TÓTH ALGORITHM

We now describe the algorithm of Balsara & Spicer (1999), later modified by Tóth (2000). The fluid quantities are evolved using the usual DS-RSA strategy, as described in §1, while the magnetic field quantities are evolved using constrained transport.

3.1 Constrained transport (CT)

CT was first developed by Evans & Hawley (1988), and uses Faraday's law to determine how the magnetic field $\boldsymbol{B}$ changes in time:

$$\frac{\partial}{\partial t}\boldsymbol{B} = -\nabla\times\boldsymbol{E}. \tag{8}$$

where $\boldsymbol{E}$ is the electric field. If take the divergence of both sides of this equation, then we get:

$$\frac{\partial}{\partial t}\nabla\cdot\boldsymbol{B} = -\nabla\cdot(\nabla\times\boldsymbol{E}) = 0. \tag{9}$$

where the second equality is true because $\nabla\cdot(\nabla\times\boldsymbol{F}) = 0$ for any vector $\boldsymbol{F}$. This means that if we evolve the magnetic field using eq. (8), then $\nabla\cdot\boldsymbol{B}$ will not change in time analytically. We can therefore maintain the divergence-free condition by simply choosing initial conditions such that $\nabla\cdot\boldsymbol{B} = 0$ (Tóth 2000). To implement eq. (8) numerically, we start by considering the x-component of the magnetic field averaged on the face $(i*)$, which we call $(bx)^{i*}$. If we define $(i*)$ as having area ΔA, then it follows that:

$$\frac{\partial}{\partial t}(bx)^{i*} = -\frac{1}{\Delta A}\int_A \nabla\times\boldsymbol{E}\cdot d\boldsymbol{A}. \tag{10}$$

Note the use of the lower case b to denote a face-averaged magnetic field quantity. (We will later use the upper-case B to distinguish the *cell*-averaged magnetic field quantities.) Then, using Stokes' theorem, we have:

$$\frac{\partial}{\partial t}(bx)^{i*} = -\frac{1}{\Delta A}\int_{\partial A} \boldsymbol{E}\cdot d\boldsymbol{r}, \tag{11}$$

where we now integrate over the path ∂A, which is the boundary that surrounds the face $(i*)$. We can discretise this equation, because we know this boundary comprises four edges in discretised space: $(i*, j*)$, $(i*, j*-1)$, $(i*, k*)$ and $(i*, k*-1)$. Furthermore, if we integrate in time, and use the fact that $\Delta A = \Delta y\Delta z$, then we obtain the following expression for $(bx)^{i*}_{n+1}$ given $(bx)^{i*}_{n}$:

$$(bx)^{i*}_{n+1} = (bx)^{i*}_{n} - \frac{\Delta t}{\Delta y}\left[(Ez)^{i*,j*}_{n*} - (Ez)^{i*,j*-1}_{n*}\right] + \frac{\Delta t}{\Delta z}\left[(Ey)^{i*,k*}_{n*} - (Ey)^{i*,k*-1}_{n*}\right], \tag{12}$$

where $(Ez)^{i*,j*}$ is the z-component of the electric field, but averaged on the *edge* $(i*, j*)$:

$$(Ez)^{i*,j*} = \frac{1}{\Delta z}\int_{(i*,j*)} \boldsymbol{E}\cdot d\mathbf{z}. \tag{13}$$

By substituting eq. (13) into eq. (12), it should be clear how the latter follows from eq. (11). Using similar reasoning, we can then derive expressions for the other two (face-averaged) components of the magnetic field, $(by)^{j*}_{n+1}$ and $(bz)^{k*}_{n+1}$:

$$(by)^{j*}_{n+1} = (by)^{j*}_{n} - \frac{\Delta t}{\Delta z}\left[(Ex)^{j*,k*}_{n*} - (Ex)^{j*,k*-1}_{n*}\right] + \frac{\Delta t}{\Delta x}\left[(Ez)^{i*,j*}_{n*} - (Ez)^{i*-1,j*}_{n*}\right], \tag{14}$$

$$(bz)^{k*}_{n+1} = (bz)^{k*}_{n} - \frac{\Delta t}{\Delta x}\left[(Ey)^{i*,k*}_{n*} - (Ey)^{i*-1,k*}_{n*}\right] + \frac{\Delta t}{\Delta y}\left[(Ex)^{j*,k*}_{n*} - (Ex)^{j*-1,k*}_{n*}\right]. \tag{15}$$

3.2 The electric field

The next step is to determine the electric field components in eqs. (12), (14)-(15). Balsara & Spicer (1999, $\varphi = 1/\sqrt{4\pi}$) use the fact that these components appear in the last three rows of the MHD flux vectors – given in eq. (5) – as fluxes of the magnetic field quantities. This means that face-averaged values of these components will be returned by the Riemann solver during the solve stage. We can convert these into the *edge*-averaged values we require by simple arithmetic averaging; to make the explanation of this clearer, we re-label the last three rows of the flux vectors as follows (where tildes signify that these are the values returned by the Riemann solver):

$$\tilde{\boldsymbol{F}}: \begin{bmatrix} 0 \\ (\tilde{F}7)^{i*}_{n*} \\ (\tilde{F}8)^{i*}_{n*} \end{bmatrix}, \quad \tilde{\boldsymbol{G}}: \begin{bmatrix} (\tilde{G}6)^{j*}_{n*} \\ 0 \\ (\tilde{G}8)^{j*}_{n*} \end{bmatrix}, \quad \tilde{\boldsymbol{H}}: \begin{bmatrix} (\tilde{H}6)^{k*}_{n*} \\ (\tilde{H}7)^{k*}_{n*} \\ 0 \end{bmatrix}. \tag{16}$$

So for example, (Ex) appears as $(\tilde{G}8)^{j*}_{n*}$ on $y =$ constant faces and as $(\tilde{H}7)^{k*}_{n*}$ on $z =$ constant faces. In our grid of cells, we know that each edge is adjacent to four faces. For example, the edge $(j*, k*)$ is adjacent to the faces $(k*)$, $(j+1, k*)$, $(j*)$ and $(j*, k+1)$, as depicted in Fig. 2a. So to obtain the x-component of the electric field averaged along this edge, $(Ex)^{j*,k*}_{n*}$, Balsara and Spicer simply take the arithmetic mean of the appropriate fluxes through the surrounding four faces (where minus signs compensate for the minus signs in eq. (5)):

$$(Ex)^{j*,k*}_{n*} = \frac{1}{4}\left[(\tilde{H}7)^{k*}_{n*} + (\tilde{H}7)^{j+1,k*}_{n*} - (\tilde{G}8)^{j*}_{n*} - (\tilde{G}8)^{j*,k+1}_{n*}\right], \tag{17}$$

and similarly for the other components:

$$(Ey)^{i*,k*}_{n*} = \frac{1}{4}\left[(\tilde{F}8)^{i*}_{n*} + (\tilde{F}8)^{i*,k+1}_{n*} - (\tilde{H}6)^{k*}_{n*} - (\tilde{H}6)^{i+1,k*}_{n*}\right], \tag{18}$$

$$(Ez)^{j*,k*}_{n*} = \frac{1}{4}\left[(\tilde{G}6)^{j*}_{n*} + (\tilde{G}6)^{i+1,j*}_{n*} - (\tilde{F}7)^{i*}_{n*} - (\tilde{F}7)^{i*,j+1}_{n*}\right], \tag{19}$$

This completes the description of the Balsara-Spicer algorithm.

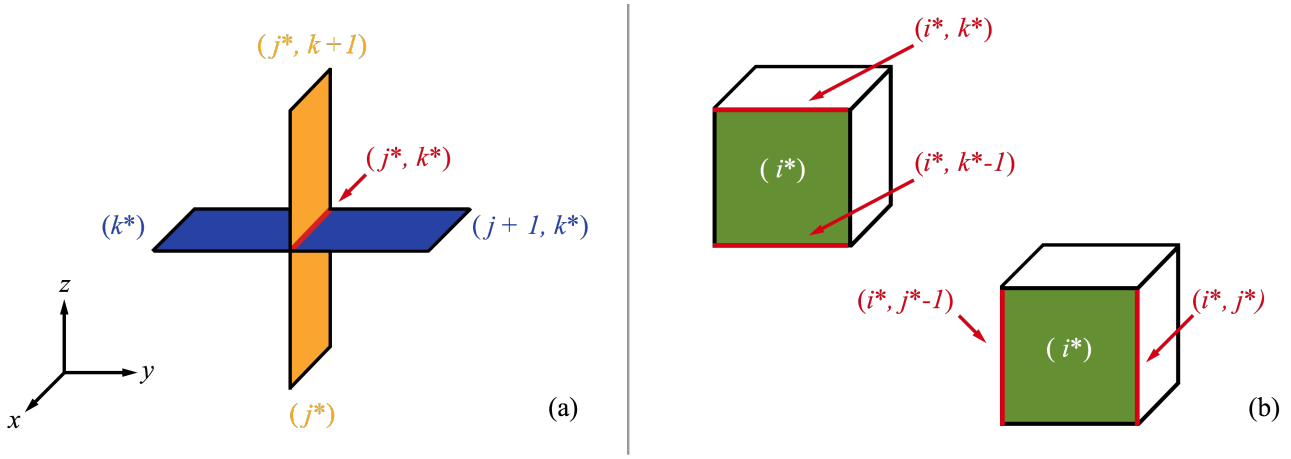


Figure 2. (a) The edge-averaged electric field. The edge $(j*, k*)$ is highlighted in red. To determine the edge average of (Ex) along this edge, we take the arithmetic mean of the *face* averages of (Ex) on the surrounding four faces. These, in turn, are returned by the Riemann solver as inter-cell fluxes during the solve stage. See §3.2 for more details. **(b) The face-averaged electric field.** We can then determine two different face-averaged components of the electric field on each face. For example, on the face $(i*)$ – highlighted in green – we can determine $(Ey)^{i*}_{\rightarrow x}$ by averaging (Ey) along the edges $(i*, k*)$ and $(i*, k*-1)$ (top left), or $(Ez)^{i*}_{\rightarrow x}$ by averaging (Ez) along the edges $(i*, j*)$ and $(i*, j*-1)$ (bottom right). See §3.3 for more details.

3.3 The cell-averaged magnetic field

It is clear that the Balsara-Spicer algorithm requires a staggered grid: the magnetic field quantities are evolved on the cell faces, while the fluid quantities are evolved at the cell centres. Whenever a cell-averaged magnetic field is required (for example, during the reconstruction stage), then we must interpolate. A linear interpolation is usually adequate:

$$(Bx)^{i}_{n} = \frac{1}{2}\left[(bx)^{i*}_{n} + (bx)^{i*-1}_{n}\right], \tag{20}$$

where we now use the upper-case B to denote a cell-averaged magnetic field. Tóth (2000) proposed a simple extension to this idea: to build the interpolation into the algorithm itself, thus eliminating the need for a staggered grid. His paper only includes the equations for a 2D system, but as he mentions, it is straightforward to extend these to three dimensions:

$$(Bx)^{\bullet}_{n+1} = (Bx)^{\bullet}_{n} - \frac{\Delta t}{\Delta y}\left[(Ez)^{j*}_{\rightarrow y} - (Ez)^{j*-1}_{\rightarrow y}\right] + \frac{\Delta t}{\Delta z}\left[(Ey)^{k*}_{\rightarrow z} - (Ey)^{k*-1}_{\rightarrow z}\right], \tag{21}$$

$$(By)^{\bullet}_{n+1} = (By)^{\bullet}_{n} - \frac{\Delta t}{\Delta z}\left[(Ex)^{k*}_{\rightarrow z} - (Ex)^{k*-1}_{\rightarrow z}\right] + \frac{\Delta t}{\Delta x}\left[(Ez)^{i*}_{\rightarrow x} - (Ez)^{i*-1}_{\rightarrow x}\right], \tag{22}$$

$$(Bz)^{\bullet}_{n+1} = (Bz)^{\bullet}_{n} - \frac{\Delta t}{\Delta x}\left[(Ey)^{i*}_{\rightarrow x} - (Ey)^{i*-1}_{\rightarrow x}\right] + \frac{\Delta t}{\Delta y}\left[(Ex)^{j*}_{\rightarrow y} - (Ex)^{j*-1}_{\rightarrow y}\right]. \tag{23}$$

which is similar to eqs. (12), (14)-(15), except we now require *face*-averaged electric field components. Note that these are *not* the same as the face-averaged components we obtained earlier from the Riemann solver. Instead, they are found by taking the arithmetic mean of the edge-averaged electric field components – of eqs. (17)-(19) – in two surrounding, parallel edges. Each face has two such face-averaged electric field components, which depend on the two edges we use to calculate this mean. For example, for the face $(i*)$, we have:

$$(Ey)^{i*}_{\rightarrow x} = \frac{1}{2}\left[(Ey)^{i*,k*} + (Ey)^{i*,k*-1}\right] \quad \text{and} \quad (Ez)^{i*}_{\rightarrow x} = \frac{1}{2}\left[(Ez)^{i*,j*} + (Ez)^{i*,j*-1}\right]. \tag{24}$$

Both of these are illustrated in Fig. 2b. We use the $\rightarrow$ subscript to distinguish, for example, between $(Ey)^{i*}_{\rightarrow x}$ through the $x =$ constant faces and and $(Ey)^{k*}_{\rightarrow z}$ through the $z =$ constant faces. Using eqs. (17)-(19), we then have:

$$(Ey)^{i*}_{\rightarrow x} = \frac{1}{8}\left[(\tilde{F}8)^{i*,k}_{n*} + (\tilde{F}8)^{i*,k+1}_{n*} - (\tilde{H}6)^{i,k*}_{n*} - (\tilde{H}6)^{i+1,k*}_{n*} + (\tilde{F}8)^{i*,k-1}_{n*} + (\tilde{F}8)^{i*,k}_{n*} - (\tilde{H}6)^{i,k*-1}_{n*} - (\tilde{H}6)^{i+1,k*-1}_{n*}\right], \tag{25}$$

$$(Ez)^{i*}_{\rightarrow x} = \frac{1}{8}\left[(\tilde{G}6)^{i,j*}_{n*} + (\tilde{G}6)^{i+1,j*}_{n*} - (\tilde{F}7)^{i*,j}_{n*} - (\tilde{F}7)^{i*,j+1}_{n*} + (\tilde{G}6)^{i,j*-1}_{n*} + (\tilde{G}6)^{i+1,j*-1}_{n*} - (\tilde{F}7)^{i*,j-1}_{n*} - (\tilde{F}7)^{i*,j}_{n*}\right], \tag{26}$$

In a similar manner, we can derive the face-averaged components through the faces $(i*)$ and $(k*)$:

$$(Ex)^{j*}_{\rightarrow y} = \frac{1}{8}\left[(\tilde{H}7)^{j,k*}_{n*} + (\tilde{H}7)^{j+1,k*}_{n*} - (\tilde{G}8)^{j*,k}_{n*} - (\tilde{G}8)^{j*,k+1}_{n*} + (\tilde{H}7)^{j,k*-1}_{n*} + (\tilde{H}7)^{j+1,k*-1}_{n*} - (\tilde{G}8)^{j*,k-1}_{n*} - (\tilde{G}8)^{j*,k}_{n*}\right], \tag{27}$$

$$(Ez)^{j*}_{\rightarrow y} = \frac{1}{8}\left[(\tilde{G}6)^{i,j*}_{n*} + (\tilde{G}6)^{i+1,j*}_{n*} - (\tilde{F}7)^{i*,j}_{n*} - (\tilde{F}7)^{i*,j+1}_{n*} + (\tilde{G}6)^{i-1,j*}_{n*} + (\tilde{G}6)^{i,j*}_{n*} - (\tilde{F}7)^{i*-1,j}_{n*} - (\tilde{F}7)^{i*-1,j+1}_{n*}\right]. \tag{28}$$

$$(Ex)^{k*}_{\rightarrow z} = \frac{1}{8}\left[(\tilde{H}7)^{j,k*}_{n*} + (\tilde{H}7)^{j+1,k*}_{n*} - (\tilde{G}8)^{j*,k}_{n*} - (\tilde{G}8)^{j*,k+1}_{n*} + (\tilde{H}7)^{j-1,k*}_{n*} + (\tilde{H}7)^{j,k*}_{n*} - (\tilde{G}8)^{j*-1,k}_{n*} - (\tilde{G}8)^{j*-1,k+1}_{n*}\right], \tag{29}$$

$$(Ey)^{k*}_{\rightarrow z} = \frac{1}{8}\left[(\tilde{F}8)^{i*,k}_{n*} + (\tilde{F}8)^{i*,k+1}_{n*} - (\tilde{H}6)^{i,k*}_{n*} - (\tilde{H}6)^{i+1,k*}_{n*} + (\tilde{F}8)^{i*-1,k}_{n*} + (\tilde{F}8)^{i*-1,k+1}_{n*} - (\tilde{H}6)^{i-1,k*}_{n*} - (\tilde{H}6)^{i,k*}_{n*}\right]. \tag{30}$$

This concludes the description of the Balsara-Spicer-Tóth algorithm.

3.4 Disadvantages of this algorithm

The simplest implementation of this algorithm involves creating six new 3D arrays, one for each of the inter-cell fluxes we require: $(\tilde{F}7)$, $(\tilde{F}8)$, $(\tilde{G}6)$, $(\tilde{G}8)$, $(\tilde{H}6)$ and $(\tilde{H}7)$. Every time we complete the solve stage for a given 1D row, we obtain the values of two of these fluxes along the entire row, and so can store them in the appropriate arrays. (The two we obtain depend on what the directional pass is.) Once a complete set of three directional passes has been completed, we can sort these arrays into another six 3D arrays, one for each of the face-averaged electric field components: $(Ex)^{j*}_{\rightarrow y}$, $(Ex)^{k*}_{\rightarrow z}$, $(Ey)^{i*}_{\rightarrow x}$, $(Ey)^{k*}_{\rightarrow z}$, $(Ez)^{i*}_{\rightarrow x}$ and $(Ez)^{j*}_{\rightarrow y}$. This is done according to eqs. (25)-(30) above. Finally, we use these components to update the magnetic field quantities using eqs. (21)-(23).

This algorithm has been shown to be reliable through standard MHD tests (Tóth 2000), but has some notable disadvantages:

(i) *Storage of information.* In a DS-RSA hydrodynamics code, we can extract a single row from the 3D grid, update all of the quantities along that row, and then return it. We do not need to store any information from that row to evolve other rows in the same directional pass. This is clearly not the case here: we need to store all of the inter-cell fluxes and electric field components.

(ii) *Processing between sets of directional passes.* In a DS-RSA hydrodynamics code, once a complete set of directional passes has been completed, the state variables require no further operations until the next set. Again, this is not the case here: we need to carry out additional operations on the inter-cell fluxes and electric field components to determine the new magnetic field quantities.

(iii) *Staggered grids.* Although the magnetic field is stored as cell averages, we still require staggered grids: the inter-cell fluxes and electric field components are stored as *face* averages. This complicates their implementation, especially with adaptive/non-uniform grids.

(iv) *Limited parallelisation.* Finally, because of the extra processing between directional passes, and the fact that the electric field components depend on inter-cell fluxes from different rows *and* directional passes, it is not immediately obvious how this code could be parallelised.

While these are arguably not major issues, they complicate the implementation of this algorithm in existing, mature astrophysical codes. We would therefore prefer an algorithm that: minimises the amount of data that needs to be stored from each row during the directional passes; reduces the amount of processing required between complete sets of directional passes; completely removes the need for any staggered grids; simplifies and maximises the possible parallelisation. We now show that such an algorithm exists, by reformulating the Balsara-Spicer-Tóth algorithm so that it is more compatible with a dimensionally-split framework.

4 A MODIFIED APPROACH

To start, we define just three new 3D arrays: $\Delta(Bx)$, $\Delta(By)$ and $\Delta(Bz)$. At the end of each complete set of three directional passes, we aim for these to contain the change in the magnetic field during those passes. To make our derivation clearer, we focus on a single cell: (i, j, k).

4.1 The x-direction pass

4.1.1 Contribution to $\Delta(Bx)^{\bullet}$

The change in $\Delta(Bx)^{\bullet}$ during each timestep is given by eq. (21), and depends on the components $(\bar{E}z)^{j*}_{\rightarrow y}$, $(\bar{E}z)^{j*-1}_{\rightarrow y}$, $(\bar{E}y)^{k*}_{\rightarrow z}$ and $(\bar{E}y)^{k*-1}_{\rightarrow z}$. We can substitute in the definition of these components, given by eqs. (30) and (28). However, we only include the $\tilde{\boldsymbol{F}}$ terms, as these are the only ones we obtain during the x-direction pass. Then, after simplification, we are left with the following expression:

$$\begin{aligned}\Delta(Bx)^{\bullet} := \Delta(Bx)^{\bullet} \quad &+\tfrac{\Delta t}{\Delta y}\tfrac{1}{8}\left[(\tilde{F}7)^{i*,j+1}_{n*} + (\tilde{F}7)^{i*-1,j+1}_{n*} - (\tilde{F}7)^{i*,j-1}_{n*} - (\tilde{F}7)^{i*-1,j-1}_{n*}\right] \\ &+\tfrac{\Delta t}{\Delta z}\tfrac{1}{8}\left[(\tilde{F}8)^{i*,k+1}_{n*} + (\tilde{F}8)^{i*-1,k+1}_{n*} - (\tilde{F}8)^{i*,k-1}_{n*} - (\tilde{F}8)^{i*-1,k-1}_{n*}\right],\end{aligned} \tag{31}$$

where use ':=' to refer to an update in the usual computational sense.

This is still not in the spirit of dimensionally-split algorithms, as the right-hand side of the expression depends on fluxes from several different rows, as shown by the different spatial indices. Instead, we want to consider what we could do after a *given* x-row – labelled (i, J, K) with fixed J, K – has been evolved. To do this, we use the fact that if $\Delta(Bx)^{\bullet}$ depends on, say, $(\tilde{F}7)^{i*,j+1}_{n*}$, then this is equivalent to saying that $\Delta(Bx)^{j-1}$ depends on $(\tilde{F}7)^{i*}_{n*}$. If we consider each term in eq. (31) in this manner, then we can rewrite it as:

$$\Delta(Bx)^{J\pm1} := \Delta(Bx)^{J\pm1} \mp \frac{\Delta t}{\Delta y}\frac{1}{8}\left[(\tilde{F}7)^{i*}_{n*} + (\tilde{F}7)^{i*-1}_{n*}\right], \tag{32}$$

$$\Delta(Bx)^{K\pm1} := \Delta(Bx)^{K\pm1} \mp \frac{\Delta t}{\Delta z}\frac{1}{8}\left[(\tilde{F}8)^{i*}_{n*} + (\tilde{F}8)^{i*-1}_{n*}\right]. \tag{33}$$

Both the right-hand sides only contain terms from the row we are currently on, $(i*)$. This means that once a given row has been evolved, we can process the contribution of the inter-cell fluxes to the magnetic field quantities immediately – using eqs. (32) and (33) – without needing to store them in any other form. It is also worth noting that these equations are near identical to the standard RSA approach given in eq. (3).

Table 2. In a dimensionally-split code, we can avoid rewriting the 1D algorithms if we make sure that the Cartesian components of the velocity are extracted (and then returned) according to this table. A similar strategy applies for the components of the magnetic field.

Dimensional pass	**x**	**y**	**z**
$(vx)^{1\mathrm{D}}$ stores	(vx)	(vy)	(vz)
$(vy)^{1\mathrm{D}}$ stores	(vy)	(vx)	(vx)
$(vz)^{1\mathrm{D}}$ stores	(vz)	(vz)	(vy)

4.1.2 Contribution to $\Delta(By)^\bullet$ and $\Delta(Bz)^\bullet$

Following a similar approach, we use eq. (22) to find the contribution to $\Delta(By)^\bullet$ during the x-direction pass:

$$\Delta(By)^\bullet := \Delta(By)^\bullet + \frac{\Delta t}{\Delta x}\frac{1}{4}\left[-(\tilde{F}7)^{i*}_{n*} + (\tilde{F}7)^{i*-1}_{n*}\right], \tag{34}$$

$$\Delta(By)^{j\pm1} := \Delta(By)^{j\pm1} + \frac{\Delta t}{\Delta x}\frac{1}{8}\left[-(\tilde{F}7)^{i*}_{n*} + (\tilde{F}7)^{i*-1}_{n*}\right], \tag{35}$$

while eq. (23) gives us the contribution to $\Delta(Bz)^\bullet$ during the x-direction pass:

$$\Delta(Bz)^\bullet := \Delta(Bz)^\bullet + \frac{\Delta t}{\Delta x}\frac{1}{4}\left[-(\tilde{F}8)^{i*}_{n*} + (\tilde{F}8)^{i*-1}_{n*}\right], \tag{36}$$

$$\Delta(Bz)^{k\pm1} := \Delta(Bz)^{k\pm1} + \frac{\Delta t}{\Delta x}\frac{1}{8}\left[-(\tilde{F}8)^{i*}_{n*} + (\tilde{F}8)^{i*-1}_{n*}\right]. \tag{37}$$

4.2 The y- and z-direction passes

We go through the same steps as above for the y-direction pass, but start by considering $\Delta(By)^\bullet$. The change in this during each timestep is given by eq. (22), and depends on the components $(\bar{E}x)^{k*}_{\to z}$, $(\bar{E}x)^{k*-1}_{\to z}$, $(\bar{E}z)^{i*}_{\to x}$ and $(\bar{E}z)^{i*-1}_{\to x}$. Again, we substitute in the definition of these components, given by eqs. (29) and (26), although this time we only include the $\tilde{\boldsymbol{G}}$ terms. After simplification, we are left with:

$$\begin{aligned}\Delta(By)^\bullet := \Delta(By)^\bullet \quad &+\tfrac{\Delta t}{\Delta x}\left[(\tilde{G}6)^{i+1,j*}_{n*} + (\tilde{G}6)^{i+1,j*-1}_{n*} - (\tilde{G}6)^{i-1,j*}_{n*} - (\tilde{G}6)^{i-1,j*-1}_{n*}\right] \\ &+\tfrac{\Delta t}{\Delta z}\left[(\tilde{G}8)^{j*,k+1}_{n*} + (\tilde{G}8)^{j*-1,k+1}_{n*} - (\tilde{G}8)^{j*,k-1}_{n*} - (\tilde{G}8)^{j*-1,k-1}_{n*}\right].\end{aligned} \tag{38}$$

Before we continue, a note on dimensionally-split codes: when we extract a given row from each full 3D array (of which there is one for each state quantity), we store them in corresponding 1D arrays. However, the quantities that are broken down into Cartesian components - the velocity and the magnetic field - must be extracted differently depending on the directional pass. For example, in the x-direction pass, (vx) is parallel to the row, and is stored in a 1D array we shall call $(vx)^{1\mathrm{D}}$. However, in the y-direction pass, it is (vy) that is parallel to the row. To avoid having to rewrite the 1D algorithms for each directional pass, we simply store (vy) in the array $(vx)^{1\mathrm{D}}$. This approach is summarised in Table 2. If we rewrite the terms in eq. (38) according to this approach – such that $\{x, F, i, 6\} \to \{y, G, j, 7\}$ and $\{y, G, j, 7\} \to \{x, F, i, 6\}$ – then we actually retrieve eq. (31). Furthermore, it can be shown that *all* of eqs. (32)-(37) can be retrieved in this manner, and for the z-direction pass as well. In other words, once the equations for the x-direction pass has been implemented, we do not need to re-write them for the y- and z-direction passes, as long as the Cartesian components are extracted and returned as described above.

4.3 Final update

Once all three passes have been completed, we simply need to perform the following operation:

$$(Bx) := (Bx) + \Delta(Bx) \tag{39}$$

and similarly for (By) and (Bz) to update the magnetic field. There are some additional subleties associated with this, regarding the pressure and energy (which depend on the magnetic field), and these are described in the next section.

4.4 Comparison with the Balsara-Spicer-Tóth Algorithm

(i) *Reduced storage of information.* We have minimised the amount of information that needs to be stored during the directional passes. Instead of storing the individual inter-cell fluxes and electric field components (which requires twelve 3D arrays), we only store the change in the magnetic field quantities (which requires just three).

(ii) *Reduced processing between sets of directional passes.* We have also minimised the number of operations required between complete sets of directional passes: we only need to perform the simple calculation given in eq. (39).

(iii) *No staggered grids.* Furthermore, the three arrays we require store only *cell*-averaged values, which means we no longer need to store any information on the cell faces. This simplifies the use of adaptive and non-uniform grids.

(iv) *Simplified parallelisation.* The modified algorithm is more easily parallelisable. For example, assume that we have a number of processors, each independently evolving some rows in a given directional pass. Each of these processors should have a copy of the $\Delta\boldsymbol{B}$ array, which are updated as described above. Then, because these all contain *changes* in the magnetic field quantities, we only require a simple 'collect'-style operation to sum over all of these copies and obtain the total change before completing eq. (39).

(v) *Dimensionally-split framework.* Finally, because this algorithm is more compatible with the dimensionally-split framework, it is considerably simpler to include in existing dimensionally-split hydrodynamics codes.

5 THE COMPLETE ALGORITHM

We will now describe the complete algorithm for each timestep. Note that we require the following sets of arrays:

(1) $7 \times \boldsymbol{U}^{\mathrm{3D}}$: the 3D arrays which store the cell-averaged state quantities as listed in eq. (4).
(2) $7 \times \boldsymbol{U}^{\mathrm{1D}}$: the 1D arrays which are used to store 1D rows extracted from the full 3D grid.

Additionally, we require two more sets of arrays for the magnetic field quantities:

(3) $3 \times \boldsymbol{B}^{\mathrm{s}}$: used to store a copy of the cell-averaged magnetic field quantities at the start of each timestep.
(4) $3 \times \Delta\boldsymbol{B}$: used to store the change in the cell-averaged magnetic field quantities during each timestep.

5.1 Before the directional passes

(a) *Make a copy of the magnetic field quantities.* Copy the three $\boldsymbol{U}^{\mathrm{3D}}$ arrays that store the cell-averaged magnetic field quantities into the corresponding $\boldsymbol{B}^{\mathrm{s}}$ arrays. These copies should not be modified during the directional passes.

5.2 The x-direction pass

(b) *Extract a 1D row.* Choose an x-direction row from the grid, which is defined as having fixed j and k indices. Extract the cell-averaged state quantities along this row from the $\boldsymbol{U}^{\mathrm{3D}}$ arrays, and store them in the corresponding $\boldsymbol{U}^{\mathrm{1D}}$ arrays.

(c) *Reconstruct the cell-averaged data.* For each $\boldsymbol{U}^{\mathrm{1D}}$ array, interpolate the cell-averaged data to obtain left and right states at each cell interface $(i*)$. These will be input states for Riemann problems. For the tests in this paper, we use the TVD scheme of Balsara (1998, $\varphi = 1\sqrt{4\pi}$), which includes a steepening algorithm. However, a popular alternative is the Piecewise Parabolic Method of Colella & Woodward (1984), an MHD version of which is summarised in §4.2.3 of Stone et al. (2008, $\varphi = 1$).

(d) Find the inter-cell fluxes. Solve the Riemann problem at each cell interface, to obtain the inter-cell flux for each state variable. We use the multi-state HLLD approximate Riemann solver of Miyoshi & Kusano (2005, $\varphi = 1$).[2]

(e) *Update the state variables.* Using the inter-cell fluxes $\tilde{\boldsymbol{F}}$ which were determined in step (d), update the cell averages stored in $\boldsymbol{U}^{\mathrm{1D}}$ using eq. (3). Note that this should be done for the magnetic field quantities as well.[3]

(f) *Return the 1D row.* Return the new cell averages stored in the $\boldsymbol{U}^{\mathrm{1D}}$ arrays to the $\boldsymbol{U}^{\mathrm{3D}}$ arrays (again, including magnetic field quantities).

(g) *Complete the x-direction pass:* repeat steps (b) to (f) for each 1D row in the x-direction (that is, for all fixed j, k).

5.3 The y- and z-directional passes

(h) *Complete all passes:* repeat the equivalents of steps (b) to (g) for the y- and z- directions. Remember to extract and return the velocity and magnetic field components in the usual dimensionally-split manner, as described in Table 2. Also note that in some code implementations, the order of the passes is alternated between timesteps to preserve the accuracy of the 1D method.

[2] A note on using non-staggered grids: in a 1D system, the parallel component of the magnetic field – which is (Bx) in the x-direction pass – should be uniform and constant in time. This automatically satisfies the divergence-free condition. The reconstruction and solver algorithms we use were designed for such 1D systems, which means that the former does not find left and right states for this parallel component in step (c), while the latter assumes it is constant across the cell interface in step (d). This is not an issue when using staggered grids, because this constant value is stored explicitly at the cell interface. However, this is clearly not the case for non-staggered grids, which means we must interpolate this parallel component to get its value across the interface. Our tests have shown that a simple linear interpolation is sufficient, for example: $(Bx)_{\mathrm{int},i*} = [(Bx)_i + (Bx)_{i+1}]/2$.

[3] This is because we require the *gas* pressure p_{G} when we return to the earlier steps. However, recall that we only store the total pressure p_{T}, so we need to calculate it by subtracting the magnetic pressure p_{M} (i.e.$p_{\mathrm{G}} = p_{\mathrm{T}} - p_{\mathrm{M}}$), which is in turn determined by the magnetic field quantities. Hence, in cases where $p_{\mathrm{M}} \gg p_{\mathrm{G}}$, not updating the magnetic field components can result in negative pressures. This issue is discussed further in Balsara & Spicer (1999).

5.4 After the dimensional passes

After all of the above steps have been completed, the fluid quantities will be fully evolved to the next timestep. We will also have three versions of the magnetic field quantities: their values at the start of the timestep (stored in $\boldsymbol{B}^{\rm s}$), their (correct) change during the timestep as determined by the constrained transport algorithm (stored in $\Delta\boldsymbol{B}$), and their values as evolved – incorrectly – through the standard RSA strategy (stored in the $\boldsymbol{U}^{\rm 3D}$ arrays corresponding to the magnetic field quantities). Then the final steps are:

(i) *Correct the energy and pressure (I).* This is an optional step. We use the magnetic field quantities stored in the $\boldsymbol{U}^{\rm 3D}$ arrays to calculate the magnetic field pressure $p_{\rm M}$ (using $p_{\rm M} = \frac{1}{2}\varphi^2\boldsymbol{B}^2$). We then subtract this contribution from the newly evolved values of the total energy (using the equation of state defined in eq. (6) as a guide) and the total pressure.

(j) *Update the magnetic field.* We replace the magnetic field quantities stored in the $\boldsymbol{U}^{\rm 3D}$ arrays with $\boldsymbol{B}^{\rm s}$ + $\Delta\boldsymbol{B}$ (in other words, their value at the start of the timestep, plus their *correct* change in value during the timestep, according to the constrained transport algorithm). Consequently, the stored magnetic field quantities now maintain the divergence-free condition.

(k) *Correct the energy and pressure (II).* If step (i) was completed: use these new magnetic field quantities to recalculate the magnetic pressure and add this back to the total energy and total pressure. This correction means that energy may not be conserved entirely (as discussed in Balsara & Spicer 1999), but avoids negative pressures if $p_{\rm M} \gg p_{\rm G}$, as described in footnote 3.

6 MHD TESTS

To illustrate the accuracy of the modified algorithm, we now present the results of four standard MHD tests. All results were obtained using the second-order TVD reconstruction of Balsara (1998), and the HLLD Riemann solver of Miyoshi & Kusano (2005). Also note that we use units of $\varphi = 1/\sqrt{4\pi}$, except for the circularly-polarised Alfvén waves, which use $\varphi = 1$.

6.1 Circularly-polarised Alfvén waves

The test of circularly polarised Alfvén waves were first described by Tóth (2000). Such waves are smooth, analytic, non-linear solutions of the MHD equations, which makes them ideal for testing the implementation of these equations. The version implemented is sinusoidal and uses a periodic domain that can fit exactly one wavelength. We test two versions: a travelling wave, where the wave returns to its original position at integer multiples of the period (which is equal to 1 in this case), and a standing wave, where it stays at its original position.

We use the initial conditions described in Tóth (2000). We start by defining a rotated coordinate system, $c_\parallel$ and $c_\perp$. These are rotated by an angle $\theta = \pi/6$ from the Cartesian x and y (and hence our discretised grid) such that:

$$c_\parallel = x\cos\theta + y\sin\theta \quad \text{and} \quad c_\perp = -x\sin\theta + y\cos\theta. \tag{40}$$

We then define the velocity and magnetic field components along these axes: $v_\parallel = 0$ for travelling waves and $v_\parallel = 1$ for standing waves; $B_\parallel = 1$; $v_\perp = B_\perp = 0.1\sin(2\pi c_\parallel)$. We then have:

(a) *Space and time.* Domain: $0 < x < 1/\cos\theta$, $0 < y < \sin\theta$; number of cells: varying from 8^2 to 32^2; total time $t = 5$.

(b) *Fluid quantities.* Across the entire grid: mass density $\rho = 1$; gas pressure $p_G = 0.1$; x-velocity $v_x = v_\parallel\cos\theta - v_\perp\sin\theta$; y-velocity $v_y = v_\parallel\sin\theta + v_\perp\cos\theta$; z-velocity $v_z = 0.1\cos(2\pi c_\parallel)$; adiabatic index $\gamma = 5/3$.

(c) *Magnetic field quantities.* The magnetic field is set up equivalently to the velocity. Across the entire grid: $\varphi B_x = B_\parallel\cos\theta - B_\perp\sin\theta$; $\varphi B_y = B_\parallel\sin\theta + B_\perp\cos\theta$; $\varphi B_z = 0.1\cos(2\pi c_\parallel)$.

The results for both the travelling and standing waves at time $t = 5$ are shown in Fig. 3, and can be compared with Figs. 8 and 9 in Tóth (2000). Note that we plot $B_\perp$, which can be determined from B_x and B_y using eq. (40). We include reference plots, which are simply the initial conditions on a grid with 128×128 cells. This was designed as a difficult test – especially at low resolutions – so the results do not coincide exactly with the expected analytic solution. However, they are comparable with results from other algorithms, as presented in Tóth (2000). It may also be possible to improve them with the use of a higher-order reconstruction.

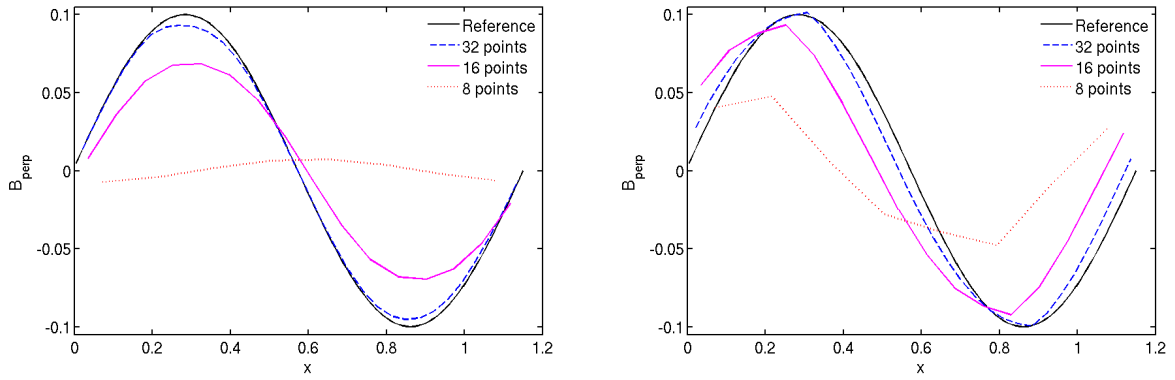
6.2 The blast problem

The blast problem is initialised as a small disk of high-pressure fluid in the centre of the grid, which then expands rapidly into the low-pressure ambient fluid. As pointed out by Balsara & Spicer (1999), it is not necessarily useful for testing whether the magnetic field is divergence free: the conditions are such that the build-up of non-zero $\nabla\cdot\boldsymbol{B}$ will not have a major effect on the overall dynamics. However, it is useful for testing the propagation of strong MHD shocks in multiple dimensions (Stone et al. 2008).

We use the initial conditions described in Stone et al. (2008) (which were in turn taken from Londrillo & del Zanna 2000):

(a) *Space and time.* Domain: $-0.5 < x < 0.5$; $-0.75 < y < 0.75$; number of cells: 300×450; total time $t = 0.2$.

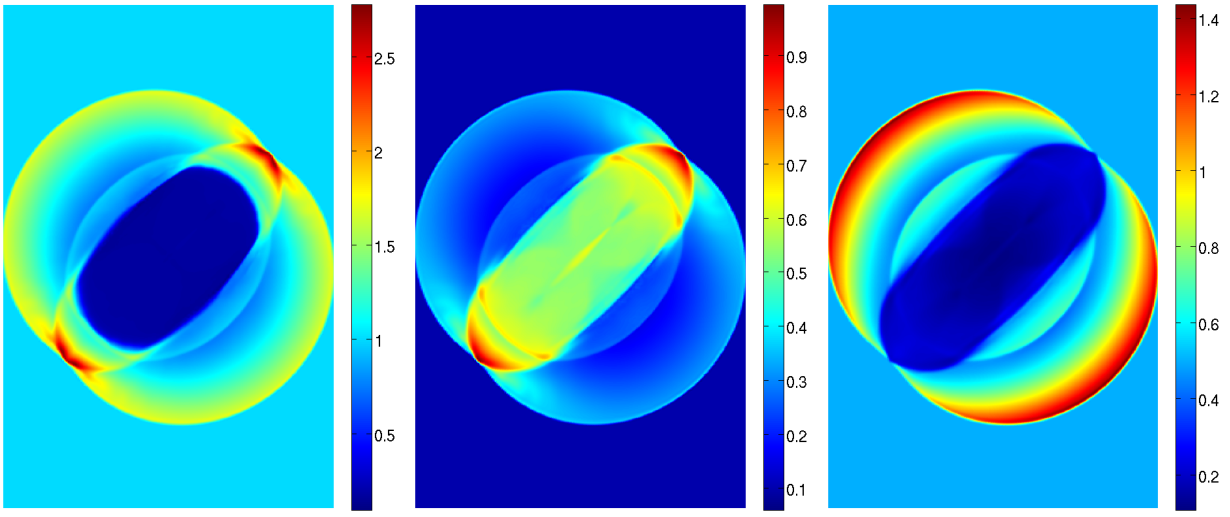
(b) *Fluid quantities.* Across the entire grid: mass density $\rho = 1$; velocity $\boldsymbol{v} = 0$; adiabatic index $\gamma = 5/3$. The gas pressure is set to $p_G = 10$ within a radius $r < 0.1$ in the centre of the grid, and $p_G = 0.1$ everywhere else.



(a) Results for travelling waves, $v_{\parallel} = 0$.

(b) Results for standing waves, $v_{\parallel} = 1$.

Figure 3. Plot of $B_{\perp} = (\sqrt{3}B_y - B_x)/2$, from the circularly-polarised Alfvén waves test at time $t = 5$, initialised as described in §6.1.



(a) Mass density.

(b) Thermal pressure.

(c) Magnetic pressure.

Figure 4. Results from the blast problem at time $t = 0.2$, initialised as described in §6.2.

(c) *Magnetic field quantities.* Across the entire grid: $\varphi B_x = \varphi B_y = 1/\sqrt{2}$; $\varphi B_z = 0$.

The mass density, thermal pressure and magnetic pressure at time $t = 0.2$ are shown in Fig. 4, and can be compared with Fig. 28 in Stone et al. (2008). The results are as expected: the blast – which would otherwise be circular - has aligned itself with the magnetic field. The near-perfect π-rotational symmetry of the results are also of note.

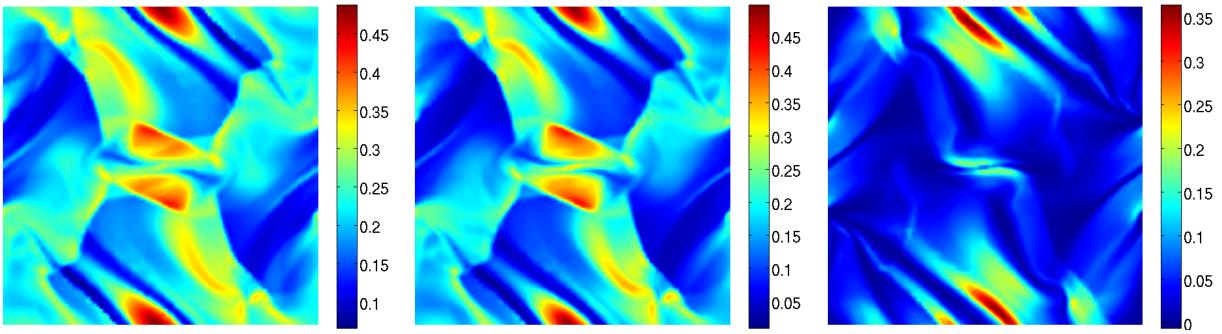
6.3 The Orszag-Tang vortex

The Orszag-Tang vortex is another commonly-used 2D MHD test, which starts with smooth initial data but becomes progressively more complex (Tóth 2000). The expected symmetry of the solution is useful for showing that the dimensionally-split algorithm (which uses the same code for each dimensional pass, but just extracts and returns the rows differently) works and has been implemented correctly.

We use the initial conditions described in Stone et al. (2008) (which were in turn taken from Orszag & Tang 1979).

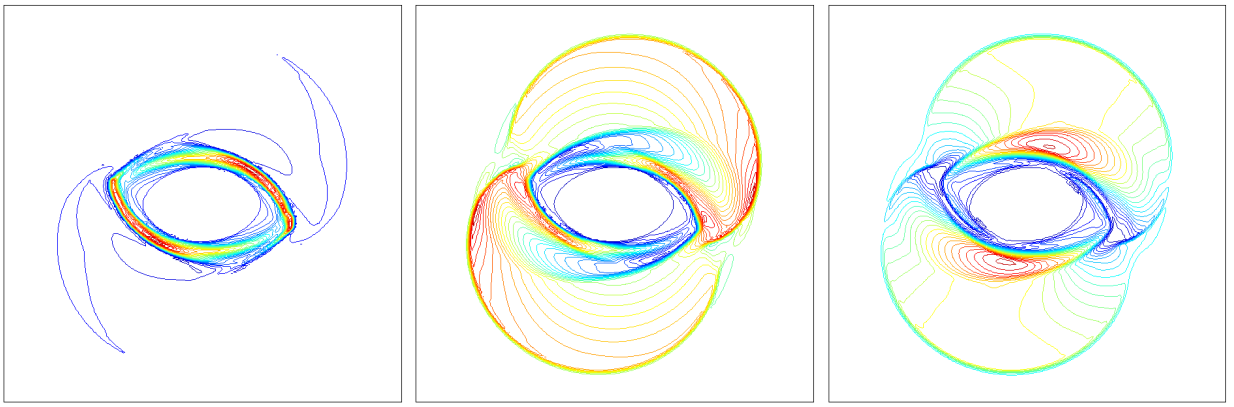
(a) *Space and time.* Domain: $0 < x, y < 1$; number of cells: 128^2; total time $t = 0.5$.
(b) *Fluid quantities.* Across the entire grid: mass density $\rho = 25/36\pi$; pressure $p = 5/12\pi$; x-velocity $v_x = -\sin(2\pi y)$; y-velocity $v_y = \sin(2\pi x)$; z-velocity $v_z = 0$; adiabatic index $\gamma = 5/3$.
(c) *Magnetic field quantities.* Across the entire grid: $\varphi B_x = -\sin(2\pi y)/\sqrt{4\pi}$; $\varphi B_y = \sin(4\pi x)/\sqrt{4\pi}$; $B_z = 0$.

The mass density, thermal pressure and magnetic pressure at time $t = 0.5$ are shown in Fig. 5, and can be compared with Fig. 22 in Stone et al. (2008). The mass density can also be compared with Fig. 16 in Tóth (2000) or Fig. 14 in Dai & Woodward (1998). The π-rotational symmetry of the results implies that our dimensionally-split approach is effective.



(a) Mass density. (b) Thermal pressure. (c) Magnetic pressure.

Figure 5. Results from the Orszag-Tang vortex at time $t = 0.5$, initialised as described in §6.3.



(a) Mass density (b) Thermal pressure. (c) Magnetic pressure.

Figure 6. Results from the rotor problem at time $t = 0.15$, initialised as described in §6.4. All contours are linearly-spaced from 0.426 (blue) to 12.4 (red) for the mass density, 0.0369 (blue) to 1.98 (red) for the thermal pressure and 0.0154 (blue) to 2.60 (red) for the magnetic pressure.

6.4 The rotor problem

Finally, we consider the rotor problem, first proposed in its current form by Balsara & Spicer (1999). This involves placing a rotating disk in the centre of a square domain, which creates rotational discontinuities and propagates Alfvén waves into the ambient fluid. The magnetic field is initialised such that it wraps around the rotor, squeezing the fluid from its original circular form. This problem is very useful for testing whether the magnetic field is divergence free, because the build-up of non-zero $\nabla \cdot \boldsymbol{B} = 0$ greatly influences the dynamics.

We use the initial conditions described in Balsara & Spicer (1999). However, as noted by Tóth (2000), the results they present are (erroneously) from a different set of conditions. We stick to what is described in the text, and compare our results with the correct results presented in other papers. We define the radius of the rotor in the centre of the grid as $r < r_0 = 0.1$. This is separated from the ambient fluid by a taper, $r_0 < r < r_1 = 0.115$. We also define $v_0 = 2$, and a constant $f = (r_1 - r)/(r_1 - r_0)$ (which is used for the taper). Then:

(a) *Space and time.* Domain: $-0.5 < x, y < 0.5$; number of cells: 256^2; total time $t = 0.15$.
(b) *Fluid quantities.* The ambient fluid: mass density $\rho = 1$; pressure $p = 1$; velocity $\boldsymbol{v} = 0$; adiabatic index $\gamma = 1.4$. The rotor is defined within a radius $r < 0.1$: mass density $\rho = 10$; x-velocity $v_x = -v_0(y - 0.5)/r_0$; y-velocity $v_y = v_0(x - 0.5)/r_0$. The taper is defined within the band $r_0 < r < r_1$: mass density $\rho = 1 + 9f$; x-velocity $v_x = -fv_0(y - 0.5)/r$; y-velocity $v_y = fv_0(x - 0.5)/r$.
(c) *Magnetic field quantities.* Across the entire grid: $B_x = 5$; $B_y = B_z = 0$.

The mass density, thermal pressure and magnetic pressure at time $t = 0.15$ are shown in Fig. 6. In each plot, there are thirty equally-spaced contours between the highest and lowest values. These results can be compared with Fig. 18 of Tóth (2000) or Fig. 25 of Stone et al. (2008), and imply that the magnetic field is divergence free. Again, it is worth noting the π-rotational symmetry of the results.

7 CONCLUSION

We have presented a partially dimensionally-split algorithm for numerically solving the equations of MHD. This was done by reformulating the algorithm of Balsara & Spicer (1999) and Tóth (2000). It is based on the standard reconstruct-solve-average strategy (using a Riemann solver), and relies on constrained transport to ensure the magnetic field remains divergence free ($\nabla \cdot \boldsymbol{B} = 0$). The dimensionally-split approach,

while not necessarily as accurate as the detailed unsplit algorithm described by Stone et al. (2008), comes with many implementational advantages over their approach, and even the Balsara-Spicer-Tóth algorithm: the information that needs to be stored during the directional passes is minimised; the processing required between complete sets of three directional passes is minimised; the need for staggered grids has been completely eliminated (simplifying the use of adapative and non-uniform grids); the algorithm is more easily parallelisable. All of these advantages also make the addition of this algorithm to existing dimensionally-split codes much simpler. This makes it particularly useful for mature astrophysical codes, which often model more complicated physical effects on top of an underlying hydrodynamics engine and so cannot be restructured easily. We included a complete description of the implementation, and illustrative source code has been made freely available online. Finally, we demonstrated the accuracy of the algorithm with several common MHD test problems.

ACKNOWLEDGMENTS

This research was supported by the Engineering and Physical Sciences Research Council, and has made use of the resources provided by the Edinburgh Compute and Data Facility (ECDF). We would also like to thank Sam Falle at the University of Leeds for his comments.

REFERENCES

Balsara D. S., 1998, Astrophys. J. Suppl. S., 116, 133
Balsara D. S., Spicer D. S., 1999, J. Comput. Phys., 149, 270
Colella P., Woodward P. R., 1984, J. Comput. Phys., 54, 174
Dai W., Woodward P. R., 1998, J. Comput. Phys., 142, 331
Evans C. R., Hawley J. F., 1988, Astrophys. J., 332, 659
LeVeque R. J., 1992, Numerical Methods for Conservation Laws. Birkhäuser
Liska R., Wendroff B., 2003, SIAM. J. Sci. Comput., 25, 995
Londrillo P., del Zanna L., 2000, Astrophys. J., 530
Mignone A., et al., 2007, Astrophys. J. Suppl. S., 170, 228
Miyoshi T., Kusano K., 2005, J. Comput. Phys., 208, 315
Orszag S. A., Tang C.-M., 1979, J. Fluid. Mech., 90, 129
Stone J. M., Gardiner T. A., Hawley J. F., Simon J. . B., 2008, Astrophys. J. Suppl. S., 178, 137
Tóth G., 2000, J. Comput. Phys., 161, 605